FEATURE – BASED DESIGN OF WESTERN SADDLE TREE SHAPES

by

V. V.S.N.S. Yugandara Rao Polisetti

Department of Mechanical Engineering

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

July, 2002

FEATURE-BASED DESIGN OF WESTERN SADDLE TREE SHAPES

A thesis submitted
in partial fulfillment of the requirements
for the degree of

Master of Technology

by

V.V.S.N.S.Yugandara Rao Polisetti

**Department of Mechanical Engineering
Indian Institute of Technology
Kanpur – 208016, India**

July, 2002

CERTIFICATE

It is certified that the work contained in the thesis entitled, "**FEATURE-BASED DESIGN OF WESTERN SADDLE TREE SHAPES**", by *Mr. V. V. S. N. S. Yugandara Rao Polisetti (Roll No. Y010558)* has been carried out under my supervision and that this work has not been submitted else where for a degree.

Dr. S. G. Dhande
Professor
Department of M.E. & C.S.E.
Indian Institute of Technology Kanpur

July 2002

Dedicated

To

My Parents

ACKNOWLEDGEMENT

I would like to express my profound, most sincere appreciation and special thanks to my guide Prof. Sanjay G. Dhande for his careful guidance and continuous encouragement during the entire period of this study. Special thanks goes to him for giving me an opportunity to work in the leading areas of CAD like Reverse Engineering, RP & RT.

I am grateful to T.V.K.Gupta, K.Siva Prasad, Puneet Tandon, and Mukul Shukla for the useful discussions I had with them regarding my thesis work. Whenever I had some trouble, either personal or academic, they were there to resolve it. My sincere thanks and regards to G. Saravana Kumar, for the technical discussions I had with him, his motivation and advice. I am very thankful to V.S.Bandaru, K.Ganesh Babu, B. Bhanu Kishore, and Suresh Behara for making the life enjoyable, apart from academics. Sincere thanks to Mr. A. Chatterjee for providing an amiable environment. I would also like to thank all the CAD Lab staff.

Finally, I wish to express my heartfelt thanks to my parents and sisters for their endless love, encouragement and endurance during my stay at IIT Kanpur.

V.V.S.N.S.Yugandara Rao Polisetti

TABLE OF CONTENTS

List of Figures

List of Tables

Nomenclature

$\mathbf{D}_{\mathrm{obj}}$	Object dimensions
$\mathbf{E}$	Set of edges representing the relationship between the vertices
$\mathbf{F}_{\mathrm{obj}}$	Constituent features of an object
$\mathbf{G}_{\mathrm{obj}}$	Feature graph accounting for object configuration
$\mathbf{Gr}$	Feature language grammar
$\mathbf{M}$	General object class
$\mathbf{N}$	A finite set of non-terminal shape elements or markers
$\mathbf{N}_f$	Neighbouring features
$\mathbf{P}$	A finite set of shape production rules
P_{ij}	Control vertices of a surface
P_{ij}'	Corrected control point for C^1 continuity between the adjacent surfaces
$\boldsymbol{P}_{end}$	End control point of a curve
$\boldsymbol{P}_{intermediate}$	Intermediate control point of a curve
$\mathbf{Q}_f$	Set of characteristic points
$\mathbf{R}$	Regular shaped object
$\mathbf{S}$	Sculptured object class
$\mathbf{S}_f$	Set of surfaces which indicate the feature form
$\mathbf{T}$	A finite set of terminal shape elements or terminals
$\mathbf{V}$	Set of vertices
F	Feature class
O	Object class
P_{ij}	Position vectors at patch corners
R	Aggregate predicate string
V_i^a	Control points of subdivided B-Spline curve
$\mathbf{a}_i(v)$ and $\mathbf{b}_i(u)$	Boundary curves
$\mathbf{n}$	Normal vector of tangent plane
b_i^r	De Casteljau points

d	Depth of saddle tree
f_i	Features or instances of feature class
g_{max}	Maximum continuity order between two adjacent surfaces
h	Deviation of the intermediate control point in consideration and the tangent plane
k	Constant
k_{max}	Maximum geometric order of surfaces
l	Length of saddle tree
o	Objects or instances of object class
u, v	Parameter values
φ_I	Feature – feature link
φ_P	Object – feature link
α_0, α_1	Bilinear blending functions

ABSTRACT

Feature-based modeling is a promising approach for computer aided geometric design of an object since it provides a systematic representation scheme for the geometric semantics of the object. It consists of two components: vocabulary and grammar. Vocabulary includes the feature and feature relationships, which is manipulated by the grammar.

In the present work, a semantic feature-based design approach has been developed for the design of western saddle tree, a class of saddle tree shapes. Using this approach, it has been found that a set of parameters can be defined for specifying the shape of this object. These parameters in turn are related to various other geometric parameters such as the location of control points, orientation of end tangents, etc., which define the features. These parameters form the vocabulary for the proposed approach. The connectivity relations among various features, both topologically and geometrically are established. These relations denote the grammatical rules. Thus a feature model is developed and verified by comparing the final shape of the saddle tree with its point cloud data. By assigning different values to the defining parameters, it is possible to get different shapes of the saddle tree. A software module is developed based on the proposed approach. This module generates western saddle of different types and sizes based on the geometric parameters, render it and further exports the CAD model in IGES format, a neutral file format for further processing. It has been tested using several case studies.

Chapter 1
Introduction

1.1 Review of Geometric Modeling & Design

The term modeling is used to mean the activity of constructing a mathematical or computer model from the description of a shape. In contrast, the term design refers to the activity of creating, in most cases interactively, a geometric shape. On the other hand, the term "geometric modeling" refers to collection of techniques or tools that may be used in both "design" and "modeling" [1]. Much of the power of contemporary geometric modeling resides in its techniques for synthesizing, allowing us to easily describe complex shapes as arrangements of simpler ones. Geometric modeling provides a description or model that is analytical, mathematical, and abstract rather than concrete. It refers to a collection of methods used to define the shape and other geometric characteristics of an object. There are three distinct aspects of geometric modeling [2]: representation, design and rendering.

1.2 Overview of Feature-based Design

Feature-based design is a form of Computer Aided Design (CAD) that allows users to describe their product as an assembly of feature instances. Broadly speaking, features are product elements that might contribute to the shape, material, tolerance, or manufacturing process specification, for example. Shape design features generally have parametrically defined geometries associated with them. The shape feature instances are established by identifying the feature's presence; its relationship with other features in the product; and assigning values to its associated parameters. Using this approach, the user is spared the struggle to describe his/her product by directly using abstract mathematics and three-dimensional geometry.

A feature can be defined as [3]:

"...a generic element of a product design, for which specific instances are defined by a set of characteristics, so that together with other features it meets the aesthetic and/or functional design requirements."

For objects having prismatic features, there are no difficulties for feature-based design as they can be easily represented using regular geometric primitives. However, feature-based design of sculptured objects is more difficult. For sculptured objects, there is no universal set of design features common to all. Instead, groups of similar objects are identified as having similar feature anatomies, and often, common product-specific shape elements and behaviours.

Fig.1.1 depicts the 'design by features' modeling approach. An object is modeled by aggregating a set of related features, which are available from the feature library. The features are manipulated according to a set of rules. Hence, there are two major components for a feature modeler to model an object – the features in the feature library and the rules for manipulating the features. With the appropriate definitions of features and the rules, a variety of objects can be modeled.

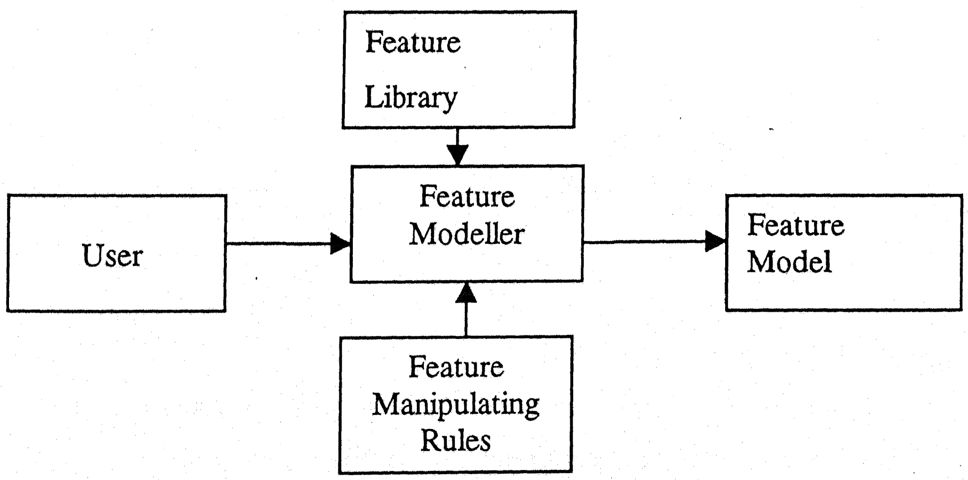


Fig.1.1 The 'Design by Features' modeling approach

1.3 Applications of Feature-based Design to free form shapes

There are numerous applications of feature-based design. As discussed in the previous section, feature-based design simplifies the modeling of the sculptured objects. Here, some of the applications of feature-based design for modeling sculptured objects are described:

1) *Garment design:* A typical application of feature-based design is human torso measurement in the garment industry. Au *et. al.* [4] describes the method of garment feature-based design. Garment panels are designed according to the measurements of a human body. Instead of using a string ruler as the traditional tailor does for measurement, a co-ordinate measuring device such as a laser scanner is used, which produces a cloud of points featuring the human torso. A mannequin is created based on this point measurement. The geometric information for garment panel design is extracted from various features of mannequin.
2) *Shoe lasts:* A shoe last is similar in shape and size to the foot intended to wear the shoe, but it is not identical. Mitchell *et. al.* [3] describes a feature-based design methodology for design of shoe lasts.
3) *Bicycle seats:* One can observe that bicycle seats are also represent a family of shapes where feature-based design find its application.

There are lots of free form shapes like artificial limbs or also other parts of human body, car bonnets, helmets, etc.

1.4 Literature Review

One approach to facilitate the feature modeling of an object is based on formal languages. Several researchers have addressed using grammar in the design process [5-6]. The discussions concentrate on how grammar facilitates the design of a regular shaped

object. Although features for sculptured object modeling are included in the feature taxonomy, it has received little attention [7]. Mitchell *et. al.* [3] discussed the feature-based approach for sculptured products. They have developed a feature model for shoe lasts. Cavendish [8] proposed a feature representation to model a sculptured object by using a transition function from the boundary of a specified secondary surface to a primary surface. Van Elsas *et. al.* [9] discussed the same approach with the transition geometry explicitly calculated.

Kim *et. al.* [10] proposed a representation formalism for feature-based design which consists of a feature algebra, a design algorithm and a system architecture/development methodology. Hoffman *et. al.* [11] proposed a procedural mechanism for generating and deploying user-defined features in a feature-based design paradigm. This study was mainly concentrated on regular shaped objects.

Au *et. al.* [12] described a semantic oriented feature language to cover the modeling of sculptured objects.

Waykole [13] has applied the feature semantics for the design of English saddle tree. The work consists of defining feature semantics and parameters for the generic English saddle tree shapes and their design. The work proves the feasibility of the approach and its implementation for obtaining surface model of generic English type saddle tree. The motivation to this work is an outcome of his work and is a natural extension in the direction of development of a feature-based approach for design of western saddle tree shapes.

1.5 Objective and Scope of Thesis

A saddle tree is a free form shape. Design of a saddle tree depends on the type of horse and the rider's requirements. Generally they are manufactured by using the judgement of experienced manufacturers. As there are a lot of variations in the types of horses and the requirements of riders, obviously, there exists separate saddle tree shape compatible to specific conditions. Hence, there is a strong need for a structured approach for saddle tree design. The same is the objective of the present thesis. This thesis

develops a feature based design of western saddle tree. The features of the saddle tree are in turn based on geometrical parameters of the object

1.6 Organization of Thesis

In Chapter 1, the overview of feature-based design and its possible applications for designing free form shapes are discussed. The work that has been carried out earlier in the area of feature-based design is also depicted. Finally the objective and scope of this work is stated.

Chapter 2 focuses on the semantics of feature-based design. The use of shape grammar in feature-based design is explained.

The proposed methodology for saddle tree shape design is thoroughly explained in Chapter 3. The relations established for designing various features are stated. The grammar rules for building the saddle tree are established.

Chapter 4 focuses on the implementation issues of the proposed methodology and data exchange format. The data structures used for representing the various components of methodology are stated. The various user interface utilities developed along with the IGES file format are also described in this chapter. Finally some results from the program are shown.

In the last chapter, conclusions derived from this work and the scopes for future work are discussed.

At the last there are 3 appendices. Appendix 1 gives the set of relations for finding the characteristic points. In Appendix 2, the relations developed for finding the curves are stated. Appendix 3 explains the method of solving the C^1 continuity problems.

Semantics of Feature Based Modeling

2.1 Background Information

In the previous chapter, some characteristics of feature-based design are explained. The basic entity in a feature model is the feature, which can also be defined as a representation of shape aspects of a product that are *mappable* to a generic shape and functionally significant for some product life-cycle phase. An essential aspect of a feature is that it has a well-defined meaning, or *semantics,* for a particular life-cycle activity.

Two important aspects of the above definition are not well covered by most current feature modeling systems. First, feature semantics is poorly defined, limiting the capability of designing the model. Second, feature semantics is poorly maintained, permitting previous design to be overruled. Such systems are said to lack *validity maintenance* facilities. In addition to lacking feature model validity maintenance, current feature modeling systems also present unexpected results after some modelling operations that are therefore said to have *ill-defined semantics* [13].

2.2 What is Semantics of Feature Based Modeling?

Feature based modeling, which is having *semantics,* is a declarative feature modeling approach. One can call it as *sematic feature modeling*. In the semantic feature modeling approach, it is essential that each feature has a well-defined meaning, or *semantics.* This is specified in *feature classes,* which are structured descriptions of all properties of a given feature type, defining a template for all its instances. Such properties include some validity conditions. All feature instances of that type should satisfy these validity conditions. These conditions, as well as the feature shape and its parameters are specified using a variety of constraint types.

In this approach, users can define their own feature classes, e.g. by inheriting from an existing feature class and adding some constraints to its definition. Feature classes are stored in feature libraries. From these feature libraries new features can be instantiated during a modeling session.

Another characteristic of semantic feature modeling is that the whole modeling process is uniformly carried out in terms of features and their entities (e.g. faces and parameters), and of constraints among these. So all modeling actions performed by the user are effectively *feature-based*, and the same applies to all output, both graphical and textual, generated by the modeling system.

2.3 Shape Grammar

A shape can be produced using shape description techniques which may be in the form of a set of instructions. Languages, either formal or natural, provide excellent basis for the study of descriptive paradigms for object representation. Shape grammars helps to capture the formalism and algebra of a shape, which can serve as a major component of feature-based design and manufacturing. Shape grammar plays a very important role in semantic feature modelling concept.

According to Longenecker *et al.* [6], a shape might be defined as a finite collection of lines, a vocabulary as a set of unique shapes, and a spatial relation (or element of a shape grammar) as the arrangement of such shapes in a particular design. Then, given some shape grammar *G*, the use of *features* as description mechanisms can be precisely characterized as being :

1. *recognizable* in sequential forms of sentences in *G*, and
2. *generable* by sentence enumeration from the grammar.

Parametric shape grammar is an extension of shape grammar in which shape descriptions become Euclidean points in space and a set of governing algebraic equations describing the relationships between these points. Parametric shape grammar can be used

to define languages of shape with proportional relationships determined in any given manner.

The use of formal rules in the design process has several distinct advantages [6]:

1. Rules are usually much less complicated than the designs they produce. They can be expressed in relatively simple terms and then easily enumerated many times by a computer.
2. Rules increase the designer's power of observation, open up new avenues of design direction and formalize thoughts and initial design requirements. This does require that these constraints be represented in an explicit and detailed form.
3. The emphasis in design is shifted away from individual designs already in existence to the general language itself. A designer can concentrate on the constructive basis of the design and an explicit awareness of its properties and structure by expressing them in specific rules.
4. Rules can be modified systematically to define new languages of design to reflect changing circumstances or to incorporate new ideas.

2.4 Use of Shape Grammar in Semantic Feature Modeling

As stated in previous section, shape grammar can be successfully used in semantic feature modelling. Au *et. al.* [12] suggested an approach to the use of shape grammar in sculptured object modeling. The same approach forms the backbone of the present work.

2.4.1 Object classification

Objects are categorized into two distinct classes, sculptured objects and regular shaped objects, based on the following set of geometric criteria [12]:

Let **M** be the object class of objects m with maximum geometric order of surface representation k_{max} and maximum continuity order between two adjacent surfaces g_{max}. Then,

$$\mathbf{M} = \{m|m = (k_{max}, g_{max}) \qquad \forall k_{max} \in \{\mathbf{I} - \{0,1\}\} \wedge \forall g_{max} \in \mathbf{I}\} \qquad (2.1)$$

Where **I** is a set of non-negative integers.

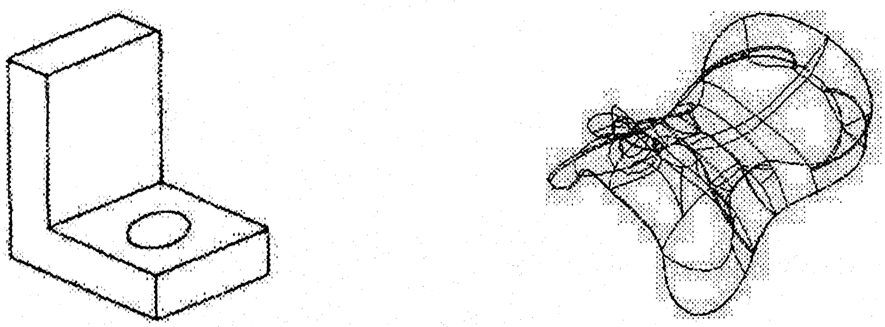
The regular shaped object class is defined as the sub-class of objects with only analytical surfaces connected by order zero or order one continuity.

$$\mathbf{R} = \{m|m = (k_{max}, g_{max}) \qquad \forall k_{max} \in \{2,3\} \wedge \forall g_{max} \in \{0,1\}\} \qquad (2.2)$$

The sculptured object sub-class is defined as the complement of **R** within the class **M**.

$$\mathbf{S} = \mathbf{M}-\mathbf{R} \text{ or } \mathbf{S} = \{m|m = (k_{max}, g_{max}) \ \forall k_{max} \in \{\mathbf{I}-\{0,1,2,3\}\} \wedge \forall g_{max} \in \mathbf{I}\} \qquad (2.3)$$

Figure.2.1 shows two typical examples of regular shaped and sculptured objects. By considering the range of the values k_{max} and g_{max}, it can be seen that for **S**, the order is much greater and the continuity requirements more stringent than that of **R**.



A regular shaped object A sculptured object

Fig.2.1 Regular shaped and sculptured object

2.4.2 Feature Language for Shape Description of Sculptured Objects

The configuration of a sculptured object is represented by a feature graph $\mathbf{G}_{obj}$

$$\mathbf{G}_{obj} = (\mathbf{V}, \mathbf{E}) \tag{2.4}$$

where $\mathbf{V}$ is a set of vertices in the graph including the object and the features, and $\mathbf{E}$ is a set of edges representing the relationship between the vertices.

The graph can be formalized by a feature language. It consists of two components: vocabulary and grammar. Both vertices and edges are vocabularies. The grammar is a set of rules, which manipulate the vocabularies.

2.4.2.1 Vocabulary of the Feature Language for Modeling Sculptured Objects

The vocabulary of the language is defined based on the feature modeling definition of a sculptured object suggested by Mitchell *et al* [3]. From the definition of feature stated in Chapter 1, the vocabularies should reflect the following main points:

- an object consists of a set of features;
- a feature is defined by a set of characteristics;
- a feature is connected to other features and
- the aesthetic and/or functional design of the object is achieved by combining the constituent features.

The vocabularies are classified into two main classes: the vertex and the edge. The vertex class consists of the following sub-classes:

(i) *O*-Object

An object is characterized by a set of its constituent features $\mathbf{F}_{obj}$ and a set of object dimensions $\mathbf{D}_{obj}$.

(ii) *F*-Feature

A feature is a constituent of an object. It includes a set of its neighbouring features $\mathbf{N_f}$ within a specific object, a set of characteristic points $\mathbf{Q_f}$ characterizing it shape, a set of surfaces $\mathbf{S_f}$ indicating the feature form, the point-surface relation and continuity constraint between adjacent surfaces within a feature.

The edge class consists of the following sub-classes:

(i) φ_P – Object – feature link

An object – feature link contains an ordered pair of an object and one of its constituent features, relation between the dimensions of a sculptured object and the characteristic points.

(ii) φ_I – Feature – feature link

A feature – feature link contains a pair of neighbouring features and the continuity constraint between the adjacent surfaces between the adjacent surfaces between two surfaces.

Fig.2.2 shows the hierarchical structure of the vocabulary classes.

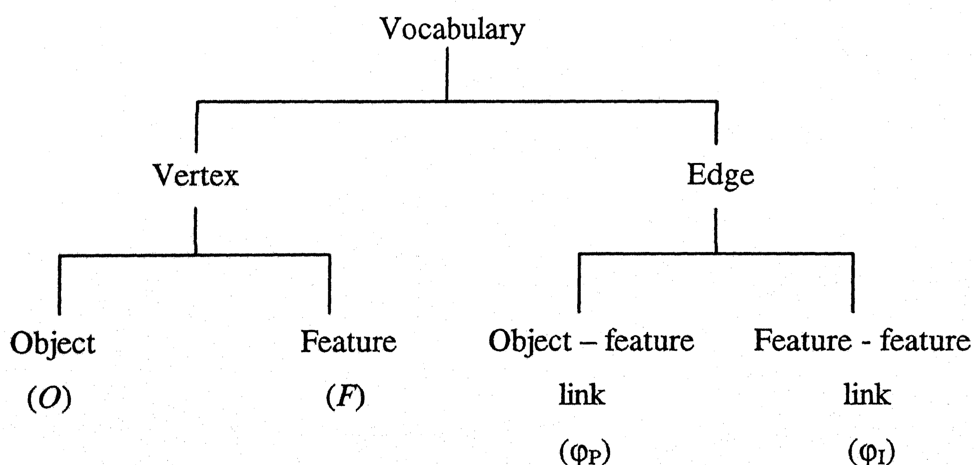


Fig.2.2 The hierarchical structure of the vocabulary classes of a feature language

The vertex vocabulary is object specific since it consists of either a constituent feature set $\mathbf{F}_{obj}$ or neighbouring feature set $\mathbf{N}_f$. This causes to restrict their usage. Therefore, the vertex vocabulary set for the complete class of sculptured objects is assumed to be very large due to large set of the sculptured object class **S**. On the other hand, the edge vocabularies stress both, configuration and geometric connectivity between vertices. They are not object specific and can be shared among the various object classes.

2.4.2.2 Grammar of the feature language for modeling sculptured objects

A feature language grammar is defined as [12]

$\mathbf{Gr} = (\mathbf{N}, \mathbf{T}, \mathbf{P}, \mathbf{G}_{obj})$

where **Gr** is the grammar, $\mathbf{G}_{obj}$ denotes the object configuration, $\mathbf{T} = \mathbf{f}_i \cup \mathbf{o} \cup \varphi_i \cup \varphi_p$ a set of terminal symbols, which are instances of the vocabulary classes. The meanings of these symbols are listed in Table 2.1, $\mathbf{N} = \{\mathbf{G}_{obj}, R, t, O, F, \varphi_P, \varphi_I\}$, is a set of non-terminal symbols. The elements of the set of terminal symbols are assigned as the instances of the elements of **N** by applying the production rules **P**, **P** is a set of production rules such that

P: { 1)	$\mathbf{G}_{obj} \rightarrow R$	: Building the sculptured object feature graph
2)	$R \rightarrow Rt \mid t$	: Defining the aggregate predicate string
3)	$t \rightarrow O\, \varphi_P\, F \mid F\, \varphi_I\, F$	: Defining predicate string
4)	$\varphi_P \rightarrow \varphi_p, \quad \forall \varphi_p \in \varphi_p$	: Instancing
5)	$\varphi_I \rightarrow \varphi_i, \quad \forall \varphi_i \in \varphi_i$	: Instancing
6)	$O \rightarrow o, \quad \forall o \in \mathbf{o}$	: Instancing
7)	$F_i \rightarrow f_k, \quad \forall f_k \in \mathbf{f}_i$	: Instancing }

The vertical bar | should be read "or".

Hence the feature graph of an object, $\mathbf{G}_{obj}$, can be written as

$$\mathbf{G}_{obj} = \sum t \tag{2.5}$$

where Σ is an aggregation symbol implying that the graph is a collection of the predicates $t = vev(\forall v \in \mathbf{V}, \forall e \in \mathbf{E})$.

Super class	**Vertex**		**Edge**	
Class	Object O	Feature F_i $(\forall i = 1,2,\ldots)$	Object – feature link φ_P	Feature – feature Link φ_I
Instance	$o_1, o_2, \ldots.$	$f_1, f_2, \ldots.$	$\varphi_{p1}, \varphi_{p2}, \ldots\ldots$	$\varphi_{i1}, \varphi_{i2}, \ldots.$
Instances set	o	f_i	φ_p	φ_I

Table 2.1 Symbols for the Vocabulary Classes and Instances

2.4.2.3 Validity of the Feature Model for Sculptured Objects by Semantic Constraints

Sculptured object modeling involves configuration and geometry. The configuration is the feature graph, which focuses the functional view of the sculptured object, while the geometry shows the sculptured object form and shape. The feature language uses grammar rules to model the sculptured object configuration first at high level of abstraction.

Let $\mathbf{G}_{obj} = (\mathbf{V}, \mathbf{E})$ be the configuration of a specific object obj with features $f_1, \ldots.,$ f_n, then,

$$\mathbf{V} = \{\ \text{obj}, f_1, \ldots., f_n\} \tag{2.6}$$

where obj, $f_1, \ldots., f_n$ are vocabularies. Hence, there exists a number of sets $\mathbf{F}_{obj}$, $\mathbf{N}_{f1}, \ldots\ldots, \mathbf{N}_{fn}$ corresponding to the vocabularies according to their definitions. The graph $\mathbf{G}_{obj}$ is valid if the following configuration constraints are fulfilled [12]:

(i) the feature set $\{f_1, \ldots\ldots., f_n\}$ of the graph $\mathbf{G}_{obj}$ conforms to the constituent feature set $\mathbf{F}_{obj}$ of the vocabulary obj. That is $\forall \mathbf{G}_{obj} = (\mathbf{V}, \mathbf{E})$, there exists a vocabulary obj $\in \mathbf{V}$ such that

$$\mathbf{V} - \{\text{obj}\} = \mathbf{F}_{\text{obj}} \tag{2.7}$$

(ii) the set of features f_j ($\forall j = 1, ...$) adjacent to a feature f_i ($\forall i \neq j$) within the graph $\mathbf{G}_{\text{obj}}$ conforms to the neighbouring feature set $\mathbf{N}_i$ of the vocabulary f_i. That is $\forall \mathbf{G}_{\text{obj}} = (\mathbf{V}, \mathbf{E})$, there exists a subgraph $\mathbf{G}_i = (\mathbf{V}_i, \mathbf{E}_i)$ where $\mathbf{V}_i = \{f_i\} \cup \mathbf{V}_i^*$

$$\mathbf{V}_i^* = \{ f_i \mid f_i \, \varphi_k \, f_j \text{ and } j \neq i, \quad \forall \varphi_k \in \mathbf{E}_i \}$$

$$\mathbf{E}_i = \{ \varphi_k \mid f_i \, \varphi_k \, f_j, \quad \forall f_i, f_j \in \mathbf{V}_i \}$$

such that

$$\mathbf{V}_i^* = \mathbf{N}_i \tag{2.8}$$

Although the configuration of a sculptured object can be determined, its form is not specified until the geometric constraints are satisfied, that is the satisfaction of

- the requirement of the object dimensions, and
- the continuity between the adjacent surfaces within the whole object

2.5 Semantic Feature Model for Sculptured Objects

A semantic feature model of an object defined by the feature language consists of three levels of abstraction in a particular domain. The three levels of abstraction are:

(i) *Object level*: this level consists of the object and the dimensions;

(ii) *Feature level*: this level consists of the features and their characteristic points. The connectivity of the features is shown by the feature–feature links. This level is related to the object level by the object–feature link. The configuration of the object is described in this level;

(iii) *Geometry level*: this level consists of the feature surfaces. Together with the configuration, the complete model of an object is described. The surfaces of a feature are

defined by the characteristic point–surface relation and the continuity between the surfaces within the feature.

Figure 2.3 shows a three–level feature model and Fig 2.4 gives a more detailed view of the model with the semantics attached to each vertex and edge.

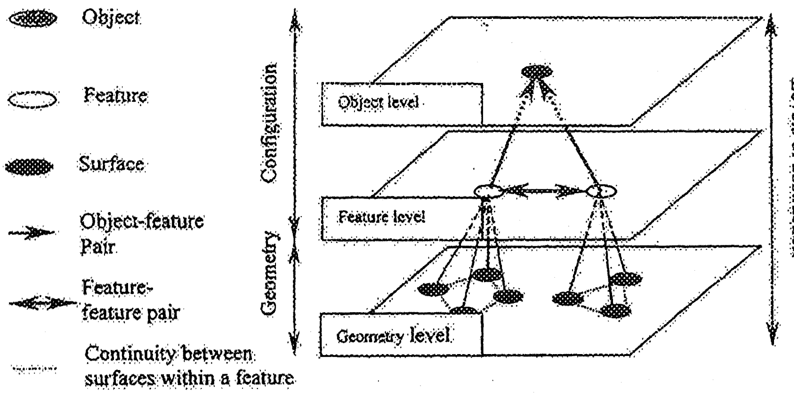


Fig.2.3 A three-level feature model

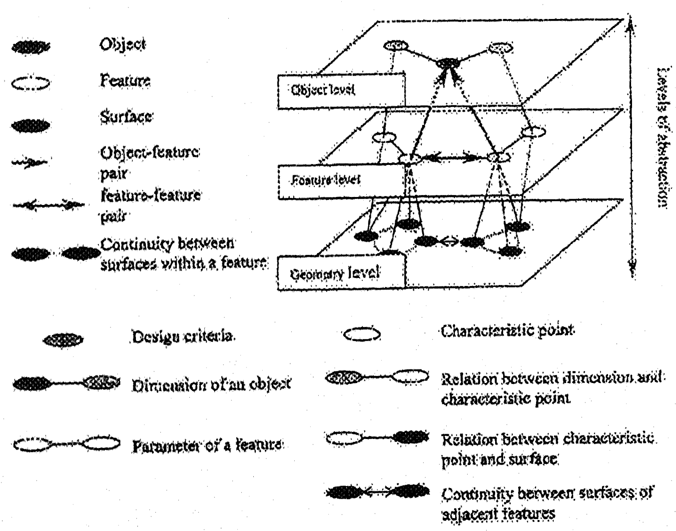


Fig.2.4. A three-level feature model with semantics

2.6 Why semantic feature modeling approach is the choice?

According to the definition of the sculptured object subclass as expressed by the relation (2.3), a saddle tree can be categorized as a sculptured object as it does not has any surface, which can be modeled using the analytical geometry principles. These surfaces can be considered as "free form" surface. There is no such well-defined method for modeling such "free form" surfaces. This is very problematic, especially in the application that employs the variant methodology for the product design. This leaves no option other than a "semantic" feature-based approach presented in this chapter. A semantic feature modeling approach helps in creating a design method, which will give a "semantic feature model" for the saddle tree. Once the "semantic feature model" for the saddle tree is available one can regenerate the saddle tree with different set of sizing parameters. Further also the manipulation of various features of the object becomes very easy.

Chapter 3

Design of Saddle Tree Shapes

A saddle is a rigid structure that connects the dynamic structures of the horse with the rider. Saddle tree is the heart of a saddle. The saddle tree is the foundation of a good saddle. Unfortunately, there are varieties of saddle tree shapes throughout the world. Some of them are shown in Fig.3.1. Every region is having a unique saddle tree type according to requirements. Besides, a variety of shapes for each type also exists.

By using the semantic approach of feature-based design, one can make a generic model for designing each of these various saddle shapes.

3.1 Basic Terminology

Before proceeding for the design of generic feature model, one should know the basic features of a saddle tree. The various features of two different types of saddle trees are shown in Fig.3.2 and Fig.3.3. Listed below are the various features of a saddle tree and related design issues which is kept in mind while designing the generic feature model.

1. *Cantle*: It is the back portion of the saddle. The term "high-backed saddle" refers to the cantle design used in the saddle tree. A higher-backed saddle can seem snugger in fit than a low one, given the seat measurement, simply because it offers more support in a higher position on the rider's back.

2. *Seat*: This is the most important part of the saddletree. Seat design is the most important factor in rider's comfort, and the rider's ability to influence the horse. A deep seat is usually more comfortable than a flat one. Some saddles have quite a sharp bucket shape to the seat. This feature really dictates the position of the rider's seat and is not favored by cross country or jumper riders. The size of the seat is not is important in a flat saddle as it is in a deep saddle. Too small a seat in a deep saddle

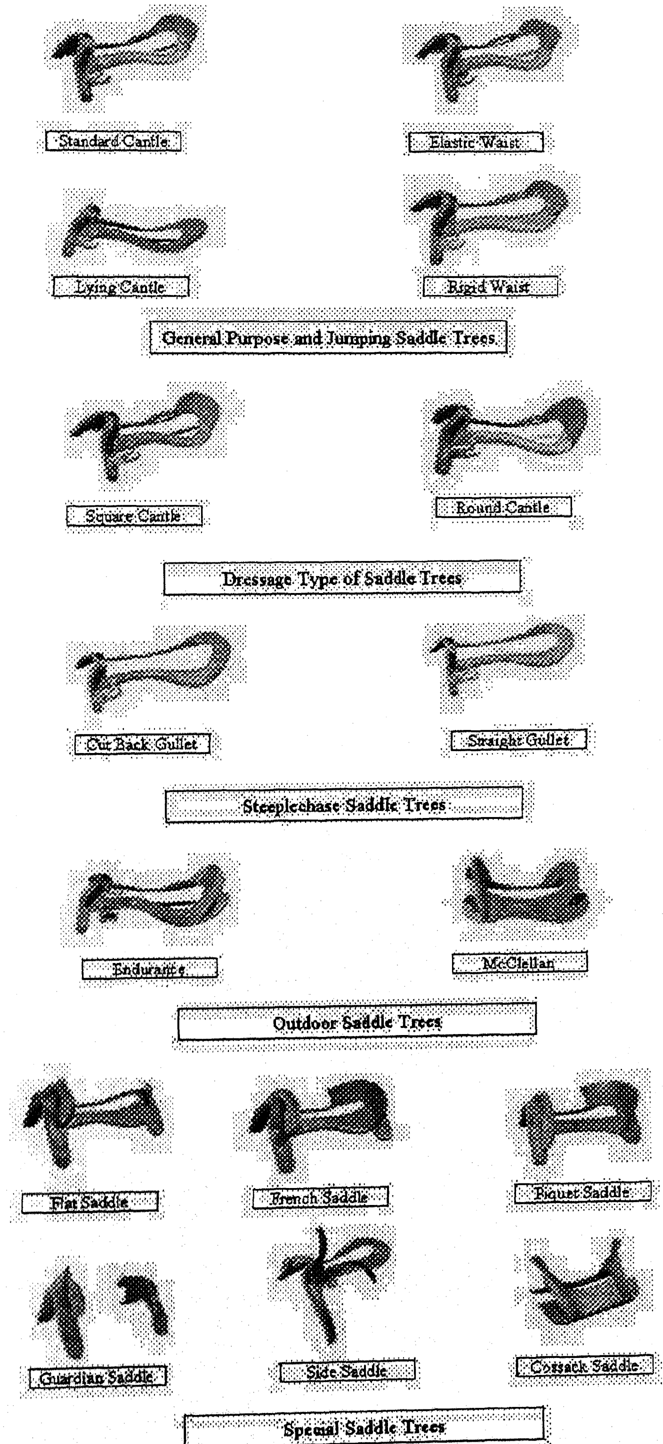


Fig. 3.1 Various Types of Saddle Trees

is uncomfortable for the rider. Again, too large a seat will give the rider very little support for their seat position.

3. *Twist*: The twist of the saddle, viewed from above, is the narrowest portion of the seat, located just behind the pommel. The twist of the seat determines how well the rider is able to get his/her legs down around the horse. That is why the twist is categorized as a separate feature. Saddles can be broadly categorized into narrow twist and broad twist, with great variation possible within each category. The general type of twist one may need depends upon the conformation of your pelvis and the way the femur is attached to it as well as the shape of the inner thigh muscle. A narrow twist is usually preferred for better influence by the rider's legs. To fit some horses, a wide tree is required for the horse's comfort and the saddle balance. A well designed saddle still has a reasonably narrow twist.

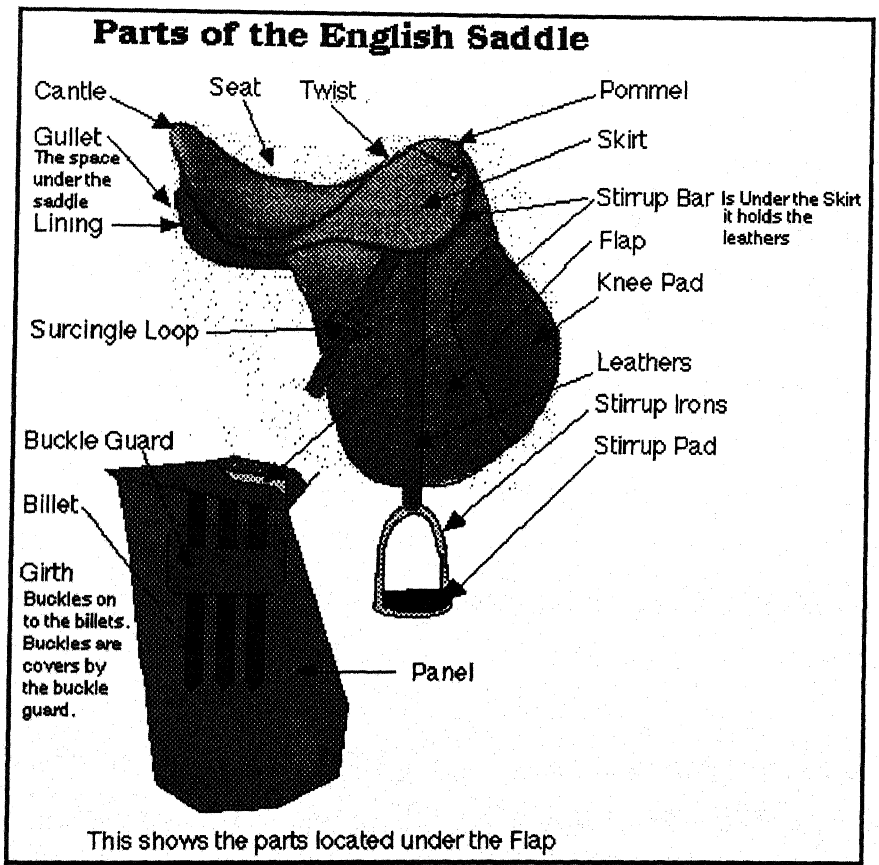


Fig. 3.2 English saddle and its features.

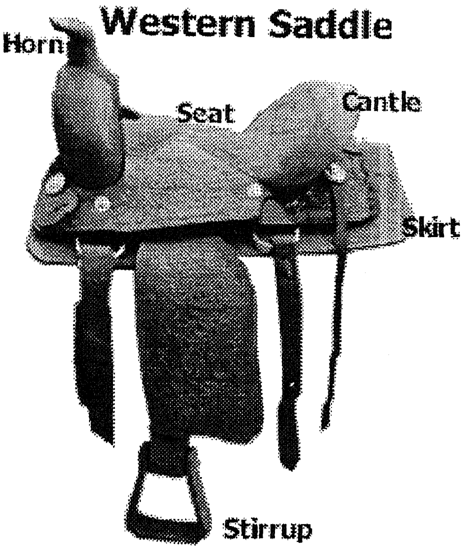


Fig. 3.3 Western saddle and its features.

4. *Pommel*: It is the front portion of the saddle.The width and height of the pommel are the design parameters. A good rule of thumb for the pommel design is: there should be approximately two to three finger's clearance between the pommel and the horse's withers. More than three fingers' clearance may mean the pommel is too high, i.e. the tree is too narrow. A saddle with less than 2-3 fingers may mean that the saddle is too wide.
5. *Legs of Saddle Tree*: The legs of the saddle (the two sides of the tree) should lie parallel to the withers, instead of sitting on top of the musculature. If the angles are too narrow, the legs will dig into the musculature, also causing the middle of the saddle to be in uneven contact with the horse's back. If they are too wide the saddle will sit down in front putting pressure on top of the withers.
6. *Horn*: This may be tall or short, thick or thin, or with a large horn cap or a small one. The angle used in attaching a horn to a tree also varies. The intended use for the saddle usually determines the horn design.

3.2 Classification of Saddle Tree Shapes

There are several different types of saddles, each used for different purpose:

1. **English Saddles**: It's the best choice of saddle for all around ridability. This types of saddle fall into 3 main groups of design:

(i) *Close Contact Saddle*: This type of saddle is used for riders who jump. It is a small, lightweight saddle, that doesn't weigh-down the horse, and allows the rider to get into a forward-seat jumping position more readily. Close contact saddles generally have fairly flat seats, when looked at from side. The cantle is usually squared off and both the pommel and the cantle are much lower that on General Purpose or Dressage types of English saddles. In a close contact saddle, the thigh of the rider is the most important part of fitting.

(ii) *Dressage Saddle*: The Dressage saddle is similar in weight and appearance as the close contact saddle, yet it has cantle rounded as opposed to the squared-off close contact saddle. Its features are deep seat and high, round cantle.

(iii) *General Purpose Saddle*: The majority of saddles sold around the world are General Purpose saddles. These saddles are suitable for flat work and jumping. This is generally preferred by all for pleasure long riding. General purpose saddles have a rounded cantle, usually a medium to deep seat for security and comfort, and have a knee roll of some kind. General Purpose saddles combine attributes of the close contact, and dressage saddle for riders who like employing both styles of riding, yet don't want the expense of two separate saddles.

2. **Western Saddles:** There are several different types of Western saddles. Western saddles differ from English saddles in some fundamental ways. Western saddles are much larger in size than English saddles. They typically are fancier in design, with lots of details. There are also many types of western saddles and all are purpose specific. There are western saddles for everything from roping, barrel racing, cutting, trail riding, hard typing, etc.

3.3 Generic Feature Model for Saddle Tree Shapes

A feature model is created based on a reference physical model of a saddle. The available physical model was western type of saddle of size 12".

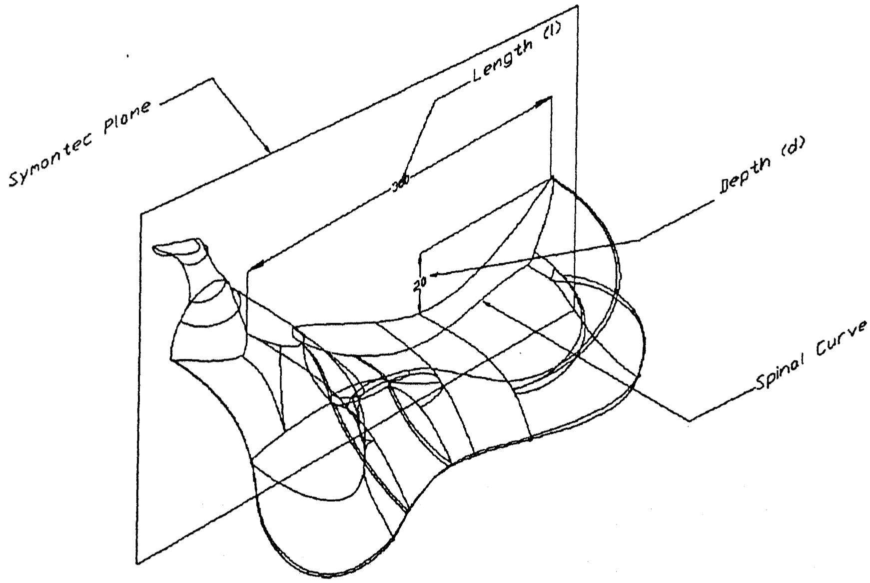


Fig. 3.4 Spinal Curve and Corresponding Characteristic Points

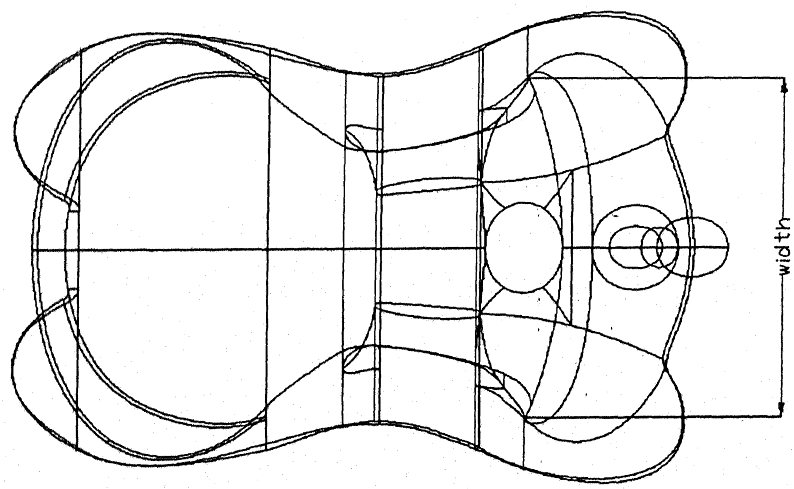


Fig. 3.5 All Curves and width dimension

3.3.1 Orientation of Saddle

For orientation of the model, there was a requirement of one plane. But there is no such plane already present in the original geometry of the objects like saddle. Still one important point to note here, is the symmetry of such objects. The same is utilized in the proposed model.

The saddle is oriented using the plane of symmetry. This plane intersects the saddle into one curve. Let us call it as "spinal curve". Fig.3.4 shows the spinal curve.

Now this spinal curve and the plane of symmetry serve the purpose of orienting the saddle in a proper position.

3.3.2 Feature Definitions

3.3.2.1 Definitions for Characteristic Points

The features are defined by a set of base vertices, called as characteristic points. These characteristic points are dependent on the sizing parameters namely length (l), depth (d), and width (w) as shown in Fig.3.4 and Fig.3.5. Every feature is having these characteristic points as their boundary vertices. These boundary vertices are as shown in Fig.3.6. Throughout the model the vertices representing the spinal curve are very important. For this case study, the proportions for the available physical model drive the locations of these vertices. All other vertices are related to these basic vertices by some proportions. All the characteristic vertices are shown in Fig. A1.1 and Fig. A1.2.

For the spinal curve, there are four characteristic points each for upper and lower side. These points are the functions of the sizing parameters, length (l) and depth (d). All points, excluding the vertices of spinal curve, are calculated as functions of the corresponding nearest appropriate spinal curve vertex. All these relations are specified in Appendix 1.

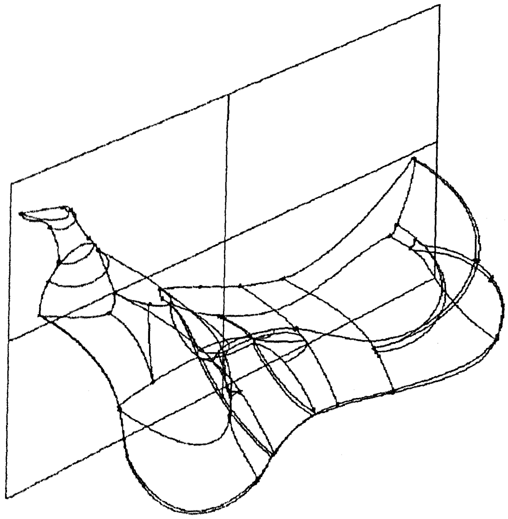


Fig.3.6 Boundary Curves and Corresponding Vertices

3.3.2.2 Calculating Boundary Curves

Using the characteristic points, the boundary curves are defined. As a whole there are a total of 267 boundary curves in the proposed model. But due to symmetry, and some of the curves generated by dividing the existing curves only 130 curves need to be calculated. All curves are cubic B-spline curves. For each boundary curves, the endpoints, which are nothing but the characteristic points, are known. The intermediate points for each curve are found using following relation:

$$P_{intermediate} = P_{end} \pm k \times dist \times v, \tag{3.1}$$

Where,

$P_{intermediate}$ = intermediate control point,

P_{end} = Nearest end control point,

k = constant,

$dist$ = three dimensional distance between the end control points,

v = direction vector.

The values of k and v differ for every curve. All these values are stated in Appendix 2.

3.3.2.3 *Surface Calculations*

Having all boundary curve definitions, the next step is to calculate the surface control points. In the proposed model, the cantle and seat surfaces are modeled using triangular patches and the rest of the features are modeled using rectangular patches.

Every triangular surface patch is considered as a degenerate case of rectangular surface patch. In this case, the common end point of any two adjacent curves is considered as the fourth boundary curve. That means the surface patch is still governed by the other three boundary curves.

As a whole there are total of 122 surface patches among which only 62 surface patches are need to be calculated. The other 62 patches are on the other side of the symmetric plane. These other half can be obtaining the calculated surface patches.

The bilinearly blended Coons patch (Fig.3.7) definition is used for finding the control net for every surface. The bilinearly blended Coons patch "control net" is obtained as

$$Net\,(u, v) = Net_1\,(u, v) + Net_2\,(u, v) - Net_3\,(u, v) \tag{3.2}$$

where,

$$Net_1\,(u, v) = \alpha_0(u)\,a_0(v) + \alpha_1(\mathrm{u})\,a_1(v),$$

$$Net_2\,(u, v) = \alpha_0(v)\,b_0(u) + \alpha_1(v)\,b_1(u),$$

$$Net_3(u, v) = \alpha_0(u)\,\alpha_0(v)\;P_{00} + \alpha_0(u)\,\alpha_1(v)\;P_{01} + \alpha_1(u)\,\alpha_0(v)\;P_{10} +$$

$$\alpha_1(u)\,\alpha_1(v)\;P_{11}$$

$\alpha_0(u) = (1 - u)$,

$\alpha_1(u) = (u)$,

$a_j(v)$, $b_j(u)$: Control points of boundary curves,

P_{ij}: position vectors at patch corners,

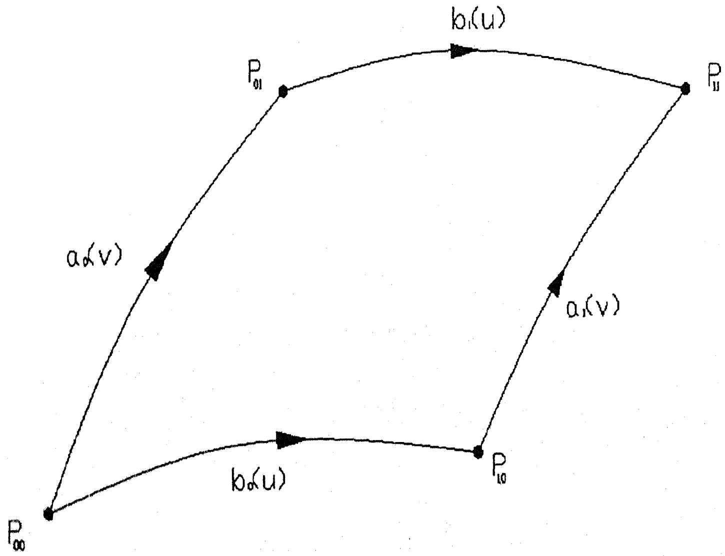


Fig.3.7 Bilinearly Blended Coons Patch

3.4 Design Methodology

3.4.1 Vocabulary Definition

The vocabulary for modelling a saddle is defined according to the saddle definition shown in Fig.3.8.

Object: Saddle. A saddle is an object vocabulary, defined by a constituent features set, F_{saddle} = {cantle, seat, and horn}, and a set of dimensions, $D_{saddle} = \{l, w, d\}$ as shown in Fig.3.4 and Fig.3.5 (features: Cantle, Seat, and Horn).

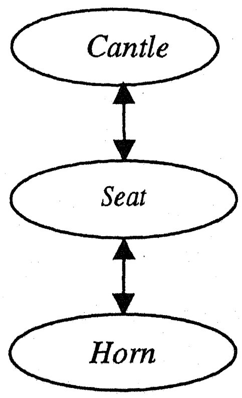


Fig. 3.8 A Saddle Configuration

The feature as a part of the object configuration is constrained by a set of neighbouring features. The geometric form of each feature is defined by a set of characteristic points Q, as the parameters and a set of surfaces S. The procedure of calculating the set of characteristic points Q and set of surfaces S is already described in Section 3.3. The relations between the points and the surfaces, and the continuity constraint between the surfaces are also included in the feature vocabulary. Table 3.1 lists the neighbouring features and the surfaces of each feature.

3.4.1.1 Object – feature link: *Part_of*

The *Part_of* vocabulary is an object – feature link signifying the configuration relation between the object and a feature. It also includes the relation between dimensions of the saddle and the characteristic points of the features. This relation is derived manually. It should be noted that this relationship is not the key to the object definition.

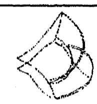
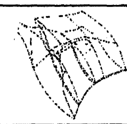
Vocabulary Classes	Neighbouring features set, N_i $\forall i = 1, 2, 3$	Feature geometry		
		Form	No. of points	No. of surfaces
Cantle(F_1)	Seat		33	34
Seat(F_2)	Cantle, Horn		17	16
Horn(F_3)	Seat		20	12

Table 3.1 The Listing of the Feature Characteristics

As explained in Section 3.3.1, the orientation of the saddle is established by finding the symmetric plane of the saddle and the corresponding spinal curve.

Let λ_l, λ_d and λ_w are the scaling proportions for the dimensions of the saddle. Denoting ($l_{desired}$, $d_{desired}$, $w_{desired}$) and ($l_{reference}$, $d_{reference}$, $w_{reference}$) as the length, depth and width dimensions of the desired and reference model, then

$$\lambda_l = \frac{l_{desired}}{l_{reference}}, \lambda_d = \frac{d_{desired}}{d_{reference}} \text{ and, } \lambda_w = \frac{w_{desired}}{w_{reference}}. \quad (3.3)$$

Then the location of a point is given by the equation,

$$\mathbf{q}_{desired} = \mathbf{\Phi}\, \mathbf{q}_{reference} \quad (3.4)$$

where,

$q_{reference}$ is the location of the point of the reference model,

$q_{desired}$ is the location of the point of the model with desired measurements and,

Φ is some function of the scaling proportions λ_l, λ_d and λ_w according to their use in calculating the point $q_{reference}$.

Therefore, when a dimension is modified, the scaling proportions λ_l, λ_d and λ_w are calculated. With these scaling proportions, the new locations of the points are calculated by Eq.(3.4) . The changes are then propagated to the geometry of the whole model.

3.4.1.2 Feature – feature link: Connect_to

The *Connect_to* vocabulary is a feature – feature link accounting for how two features are connected geometrically. It consists of a set of order one continuity constraints between two surfaces.

The continuity constraint between two surfaces s^h and s^{h+1} is expressed as follows:

(i) G^0 continuity

$$p_{0j}^{h} = p_{j0}^{h+1}, \qquad \forall j = 0, 1, \ldots, k \qquad (3.5)$$

where p_{ij}^{h} is the control point of patch h at row i and column j.

(ii) G^1 continuity

For G^1 continuity between the adjacent patches, it is ensured that the tangent planes of the two patches, along the common boundary curve, are coplanar. The details of this approach are explained in Appendix 3.

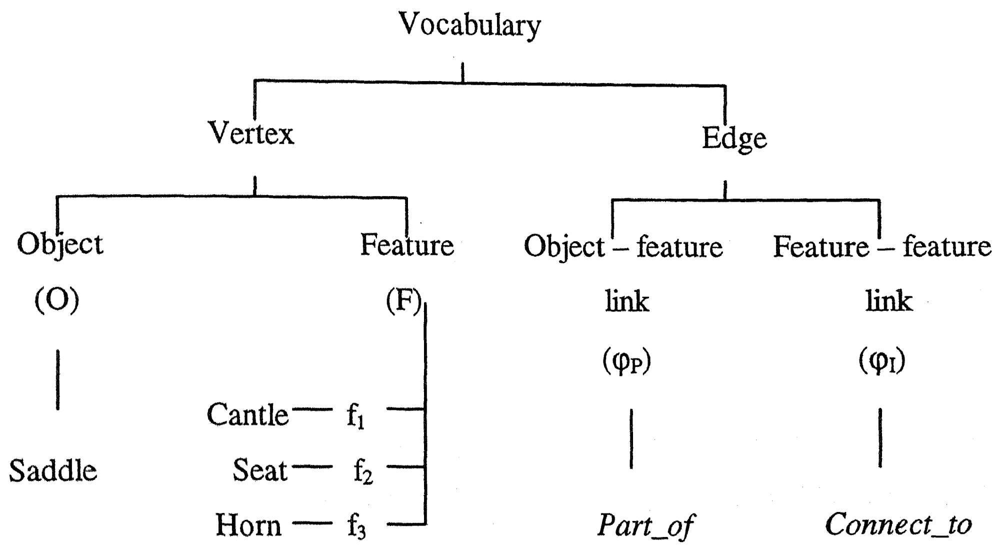


Fig.3.9 The hierarchical structure of the vocabulary classes of the feature language for a saddle

These two constraints are also applied to two adjacent surfaces within a feature. Fig.3.9 shows the hierarchy of the vocabularies for modelling a saddle.

3.4.2 Feature-based grammar for saddle definition

The grammar Gr for modeling a saddle is rewritten as

$Gr = (N, T, P, G_{saddle})$

Where,

$N = \{G_{saddle}, R, t, O, F, \varphi_P, \varphi_I\}$,

T = {Saddle, Cantle, Seat, Horn, *Part_of*, *Connect_to*},

P is a set of production rules such that

P: { 1) $\mathbf{G}_{saddle} \rightarrow R$: Building a saddle feature graph

2) $R \rightarrow Rt \mid t$: Defining the aggregate predicate string

3) $t \rightarrow O\ \varphi_P\ F \mid F\ \varphi_I\ F$: Defining predicate string

4) $O \rightarrow$ Saddle : Instancing

5) $F_i \rightarrow$ Cantle|Seat|
Horn : Instancing

6) $\varphi_P \rightarrow$ *Part_of* : Instancing

7) $\varphi_I \rightarrow$ *Connect_to* : Instancing }

3.4.3 Building the saddle

A specific feature graph unique to the saddle is built according to the configuration semantic of the vocabularies as shown in fig 3.9. Once the configuration of the saddle is defined, the geometric connectivity is set up. The feature graph $\mathbf{G}_{saddle}$ description is as follows:

Saddle *Part_of* Cantle

Saddle *Part_of* Seat

Saddle *Part_of* Horn

Cantle *Connect_to* Seat

Seat *Connect_to* Horn

A saddle feature model can be generated with the defined feature graph when the dimensions of the saddle are specified. Fig.3.10 shows a shaded image of a saddle tree.

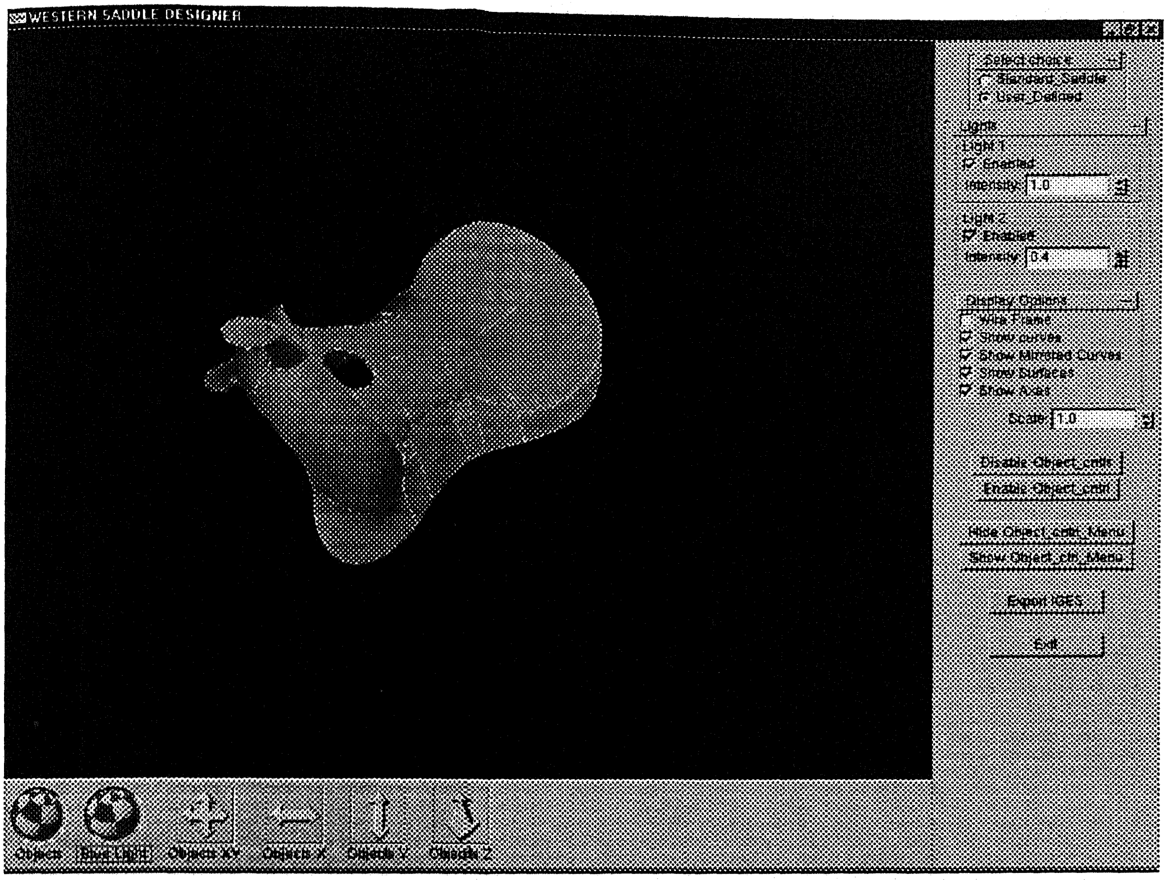


Fig.3.10 Shaded model of western saddle tree (Output of Saddle Designer)

Implementation of Design Methodology

For checking the validity of the proposed methodology, one software called "Western Saddle Designer" is developed. Fig. 4.1 shows the look of the software window. The software is developed using C++. OpenGl and GLUT libraries are used for rendering the model and GLUI is used for creating the Graphics User Interface (GUI). The following sections discuss the various software details.

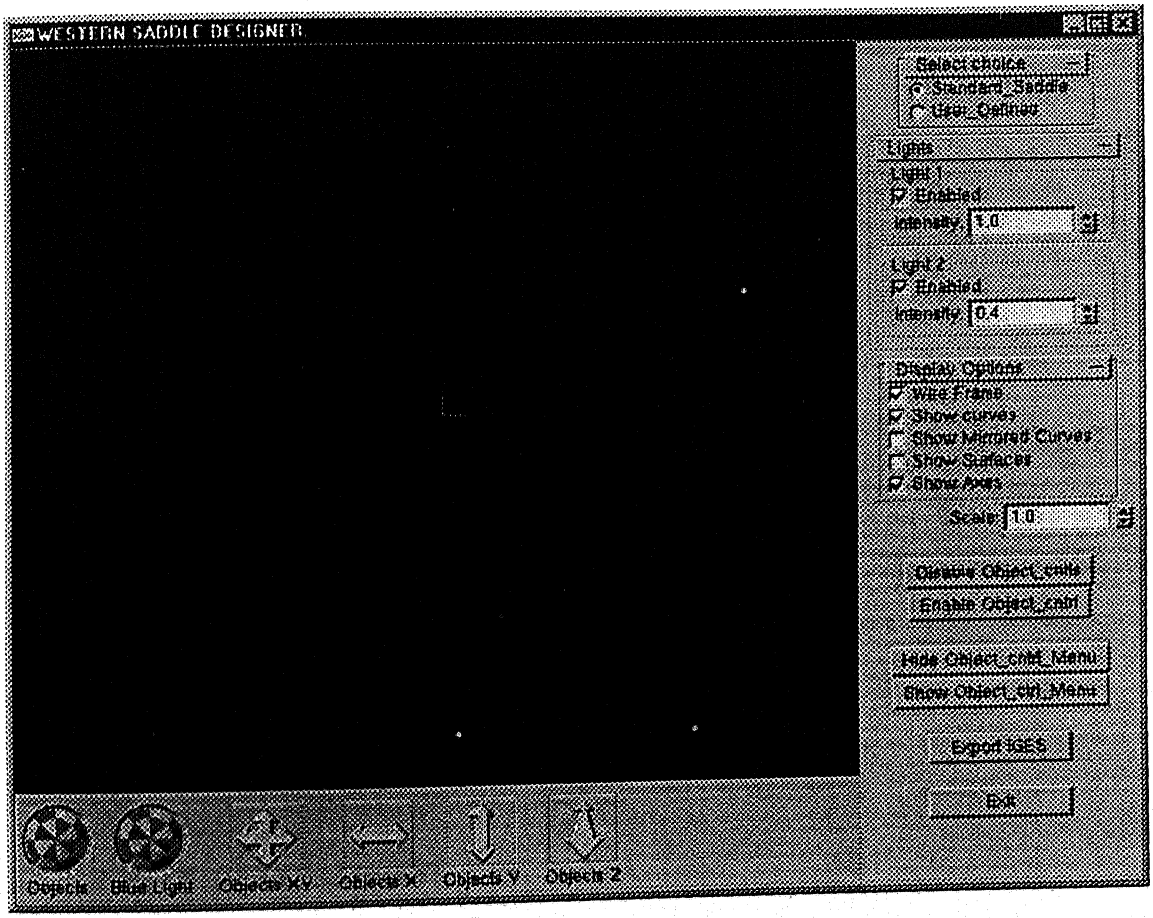


Fig 4.1 Western Saddle Designer

4.1 Structure of Design Software

The data structures used in making the software are discussed in this section.

1) *Curve*: Initially four control points represent a curve. But for solving the problems of continuity some curves are represented by six control points, which are found using the OSLO algorithm specified in Appendix 2. Hence the data structure for the curve is as follows:

```
struct data_curve
{
   float ctrlpoints[6][3];
};
```

2) *Surface*: Every surface is constructed using four boundary curves. All surfaces are cubic, having 6×6, 6×4 or 4×6, and 4×4 control points depending on the control points of the boundary curves are constructed. The variable key is used to specify the order of the control points of the surface patch, which was used while exporting the surface in IGES format and also while rendering. For that reason, the data structure for the surface is:

```
struct sface_data
{
int key;
union
{
  float cpts44[4][4][3];
  float cpts64[6][4][3];
  float cpts46[4][6][3];
  float cpts66[6][6][3];
}scpts;};
```

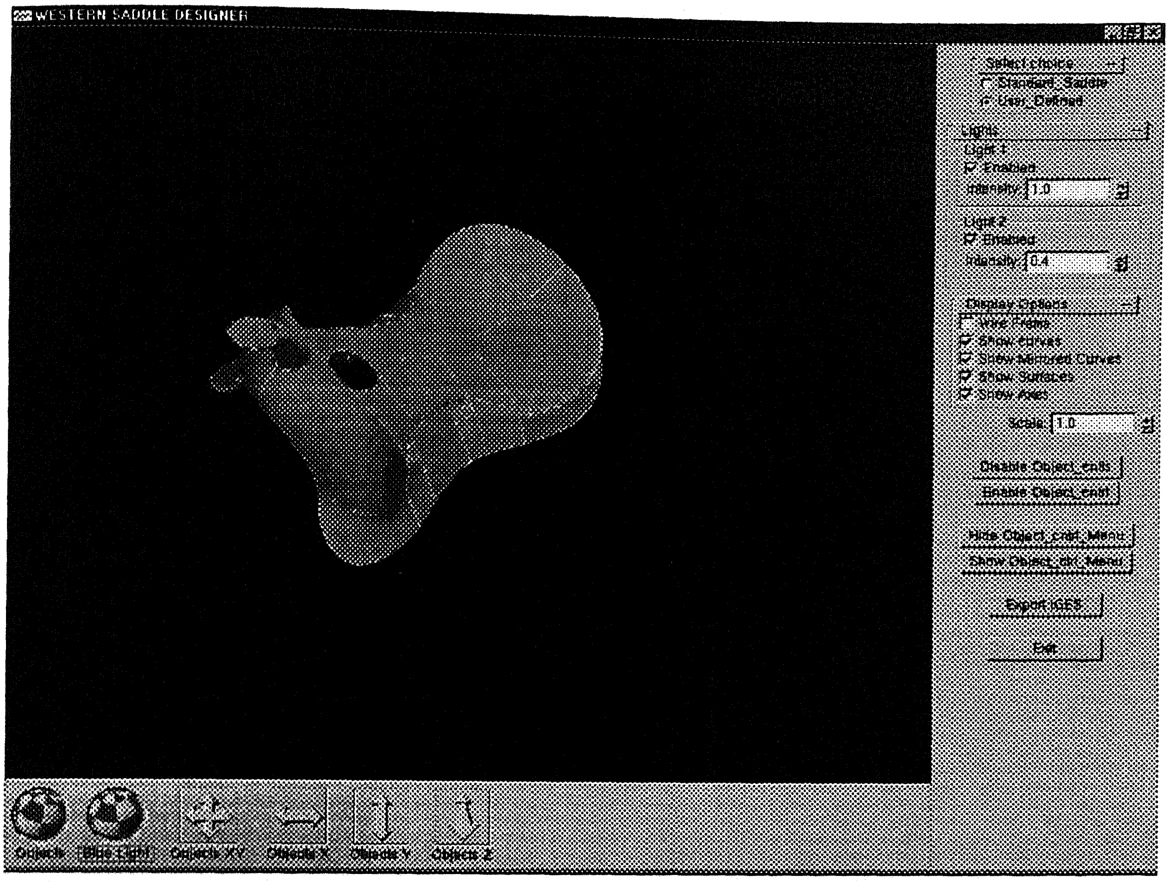


Fig. 4.2 "Western Saddle Designer" in Action

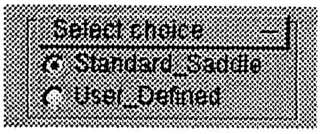
3) *Feature*: A feature is constructed by a set of surfaces. Hence the data structure for a feature is as follows:

```
typedef struct feature
{
  surface srf[14];
}ff;
```

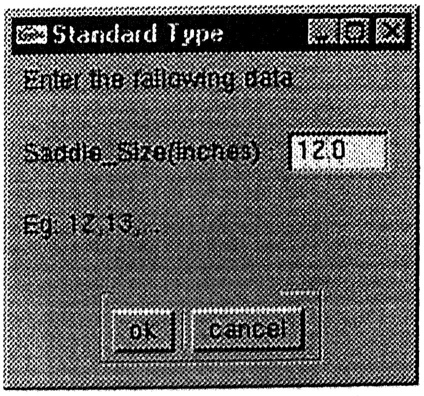
4.2 User Interface

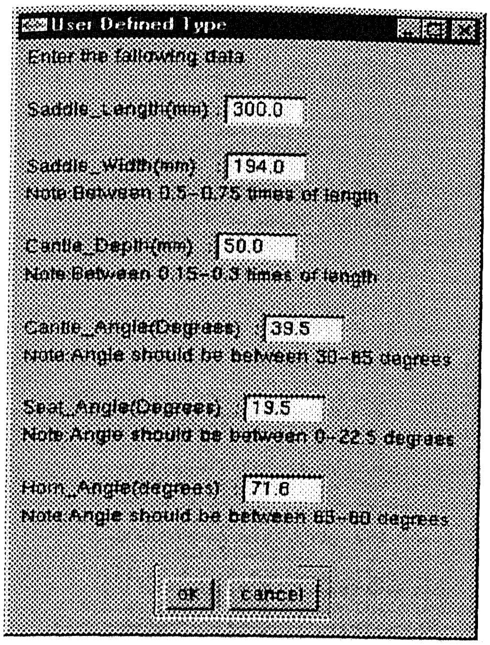
A user has to select appropriate things from the controls and tool bars provided in the main software window it self. The following section will give the brief introduction about the different controls and tool bars used in the software.

4.2.1 Choice Selection Panel

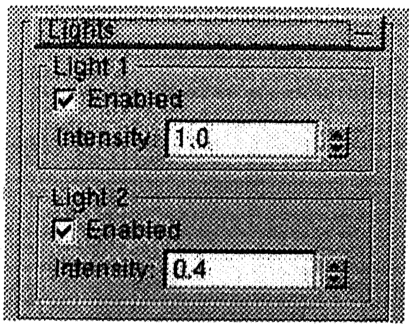


Selection panel consists of two radio buttons one for standard type saddle generation, and other for user defined. When standard type radio button is clicked standard type sub window will be opened and it asks for the size of the saddle in inches as shown below. Similarly when you clicked on user-defined type, a user defined type sub window will be opened and asks for the length, width, cantle depth, cantle angle, seat angle, and dome angle as shown below.



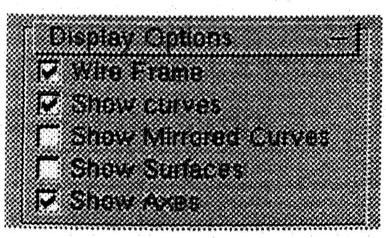


4.2.2 Lighting Panel



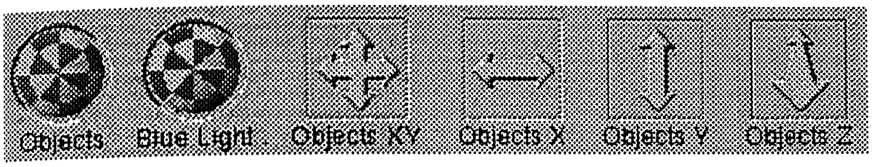
In the software two lights have been provided situated at two points. Users can enable/disable one or both of the lights or change the intensity of both the lights using these controls.

4.2.3. Display Options Panel



This panel is used for displaying the different entities like curve, surfaces, and wire frame model of the saddle.

4.2.4. Viewing Controls



These controls are used for the dynamic viewing of model. This panel provides the following controls

1.Objects: Used for the rotation of the object

2.Blue Light: Used for the rotation of the blue light source.

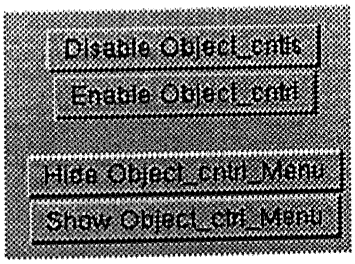
3.Objects XY: Used for the translation along both X & Y axes.

4.Object X: Used for the translation along X axis only.

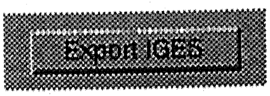
5.Object Y: Used for the translation along Y axis only.

6.Object Z: Used for Zoom In/Out of object

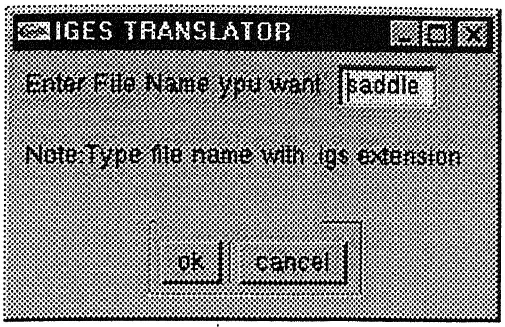
For enabling, disabling, hiding, and showing of these viewing controls fallowing buttons are created.



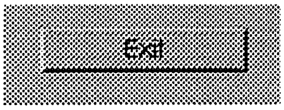
4.2.5 Export IGES button



This button is used for creating an IGES file of the model developed. When this button is clicked, a window will be opened and asks for the file name to export the model data in IGES format.



4.2.6 Exit Button



This button is used for exiting the software.

4.3 Graphics Utilities

For developing the software, various OpenGL functions are used for displaying the required information. The surfaces are displayed by using the following function for displaying nurbs:

```
void gluNurbsSurface(GLUnurbsObj *nObj, GLint uknotCount,
GLfloat *uknot, GLint vknotCount, GLfloat *vknot, GLint
uStride, GLint vStride, GLfloat *ctlArray, GLint uorder,
GLint vorder, GLenum type);
```

For displaying the curves the following function is used:

```
void gluNurbsCurve(GLUnurbsObj *nObj, GLint knotCount,
GLfloat *knot, GLint stride, GLfloat *ctlArray, GLint
order, GLenum type);
```

All the parameters in the above functions are self-explanatory.

4.4 Computational Results

The software developed for building the saddle tree is tested for various cases. These cases are explained in the next three subsections.

4.4.1 Software Validation

Before proceeding for any design results, it is necessary to check the developed software. The developed software, Western Saddle Designer, is having provision for data transfer, the model developed is exported from the software and point cloud is generated from this model and compared with the existing saddle point cloud. Figure 4.3 shows the error plot between the point cloud data generated from the "Western Saddle Designer" and the available point cloud of the 12" western Saddle tree. The results obtained are quite encouraging. Although the maximum error is about 11.8 mm, but a major portion of the lies with error less than 1.8mm.

4.4.2 Change of Size

The various features of a western saddle are discussed in Section 3.2. In fact, this work is carried out using the geometric information of General Purpose western saddle only. The commonly used sizes of western saddle vary in the range 12" to 17", in the steps of 0.5". For the purpose of illustration, the saddle tree of 17" size is modeled using "Western Saddle Designer" and examined in various views. Figure 4.4 presents one the view of the model, and Fig 4.5 shows the rendering of the after being exported in neutral file format.

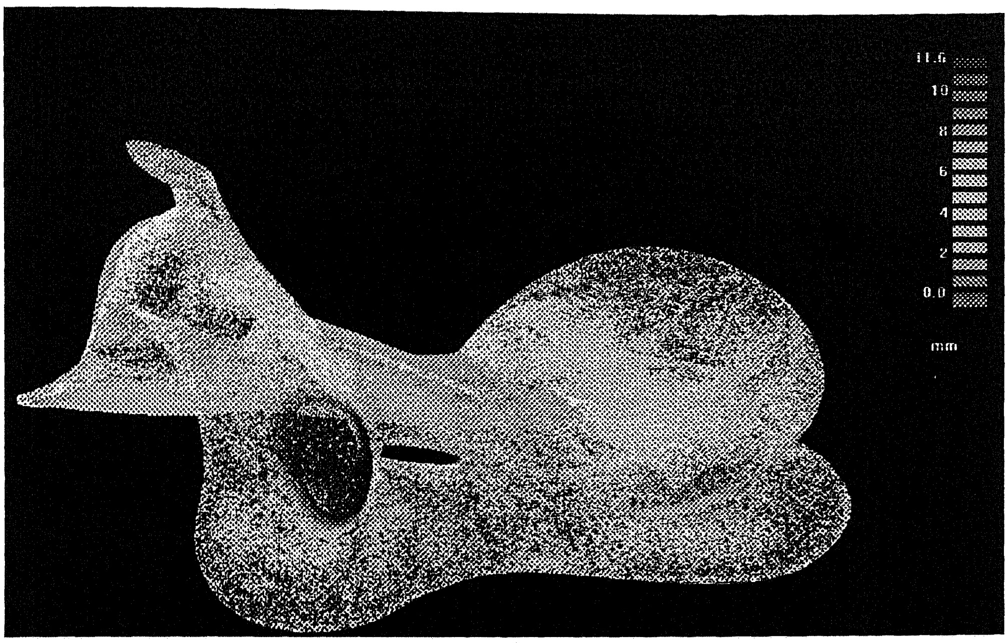


Fig. 4.3 Error plot between the point cloud data generated from the "Western Saddle Designer" and the available point cloud of the 17" General Purpose Saddle tree

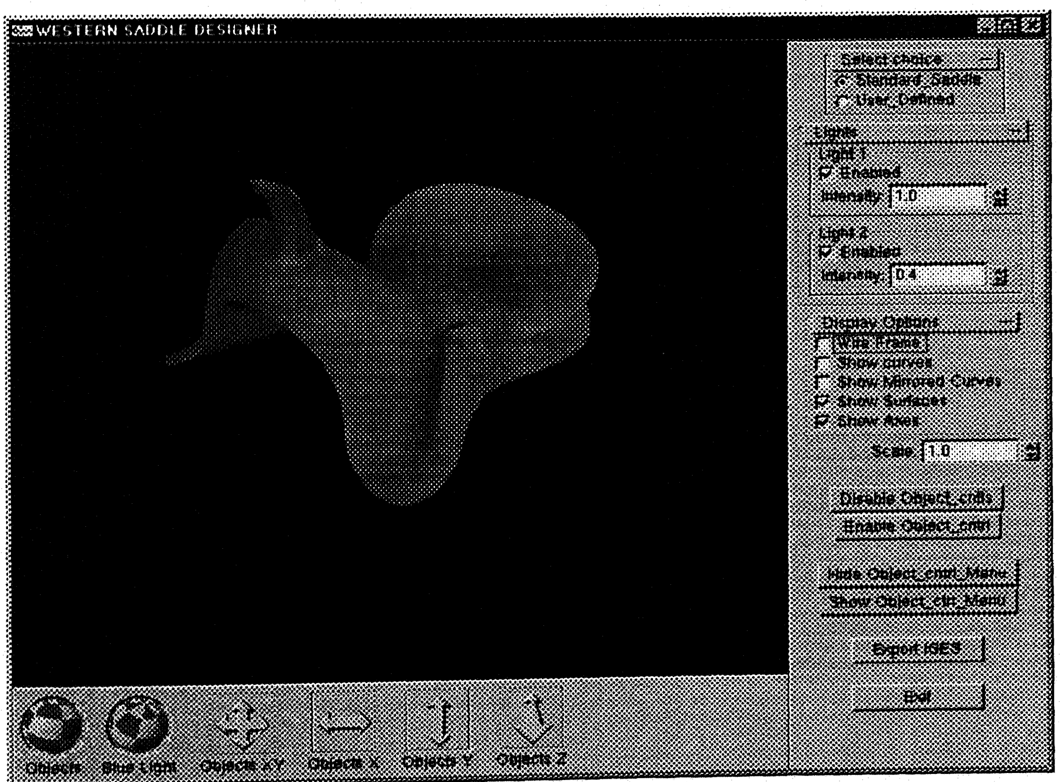


Fig. 4. 4 Result – 17" Western Saddle Tree

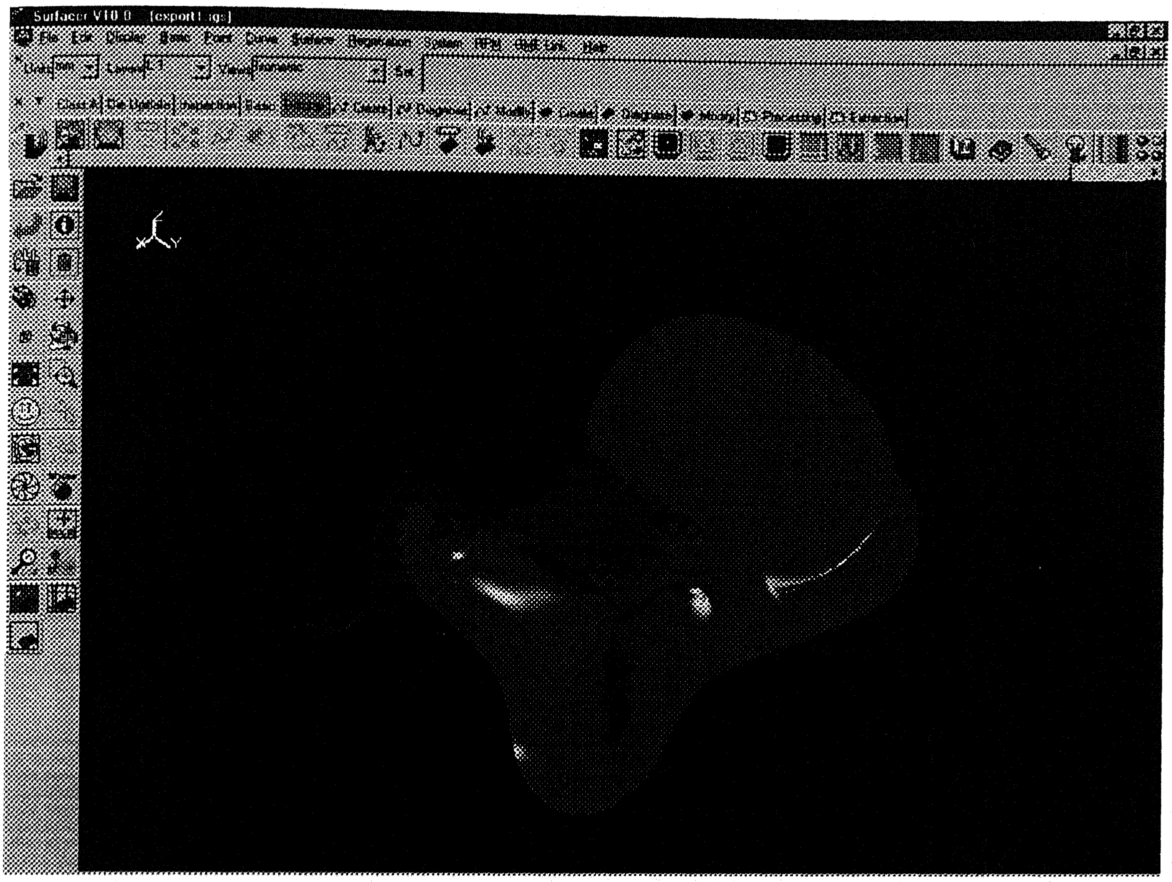


Fig.4.5 Result – exported model of 17" Saddle Tree

4.4.3 Feature-based Modification

Using the user_defined radio button in the menu of the "Western Saddle Designer" i.e. designing own saddle tree, user can do various modifications in a design already available to the library. According to the provided options, a user can manipulate the cantle shape by giving the angle of the spinal-cantle curve endpoint vector with horizontal X-axis and by stating the percentage increase/decrease in the length of the spinal-cantle curve. For manipulating the seat, a user has to input the percentage increase/decrease in depth of the seat. The pommel region can be modified by specifying the width (*w*) (measurement procedure illustrated in Fig.3.4) as width is the major shape deciding factor in pommel design. At the last the "Western Saddle Designer" queries the user to specify the size of his "customized" saddle tree.

examples of such user customization is illustrated here:

1) *Example 1:* Figure 4.6 and Fig 4.7 shows one example of a "user customized saddle tree" for the following set of input parameters:

Manipulate Cantle:

Angle = 60 degrees,

Manipulate Seat Angle:

Angle = 15 degrees,

Manipulate Dome Angle:

Dome Angle= 75 degrees,

Manipulate length, width, and depth:

Saddle length = 400mm,

Saddle width = 260mm,

Cantle depth = 75 mm.

2) *Example 2:* Figure 4.8 and 4.9 shows one example of a "user customized saddle tree" for the following set of input parameters:

Manipulate Cantle:

Angle = 65 degrees,

Manipulate Seat Angle:

Angle = 10.5 degrees,

Manipulate Dome Angle:

Dome Angle= 70 degrees,

Manipulate length, width, and depth:

Saddle length = 350mm,

Saddle width = 235mm,

Cantle depth = 60 mm.

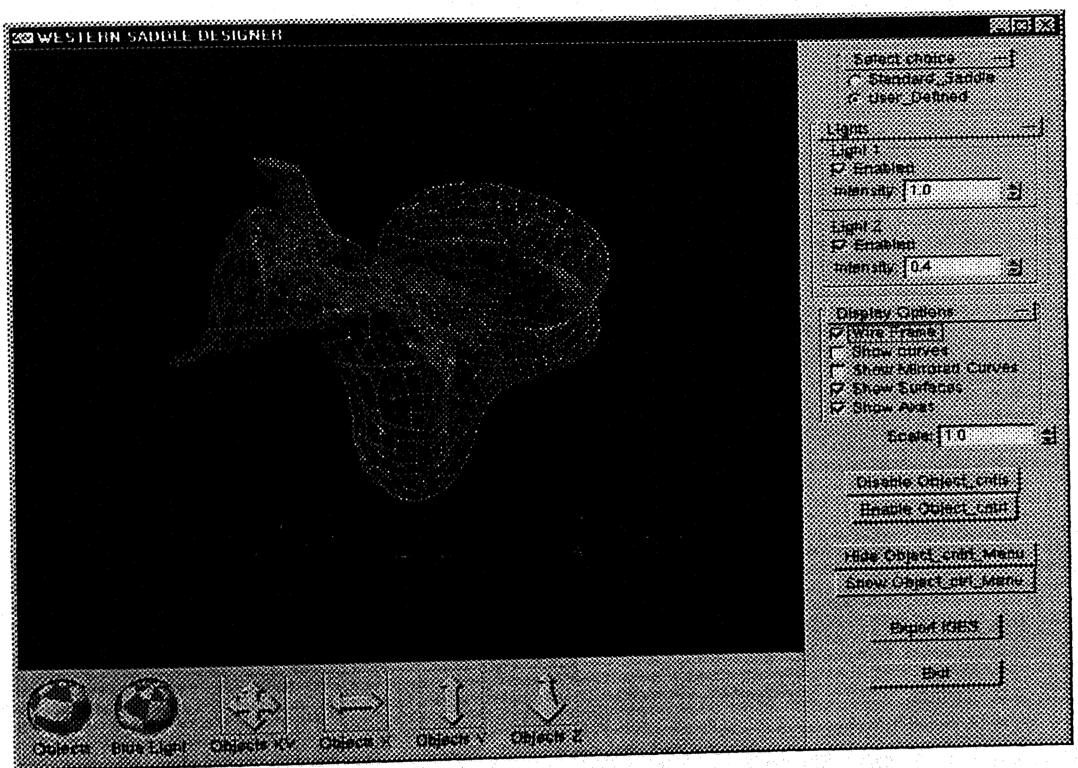


Fig 4.6 User Customization Example1 (Wire Frame Model)

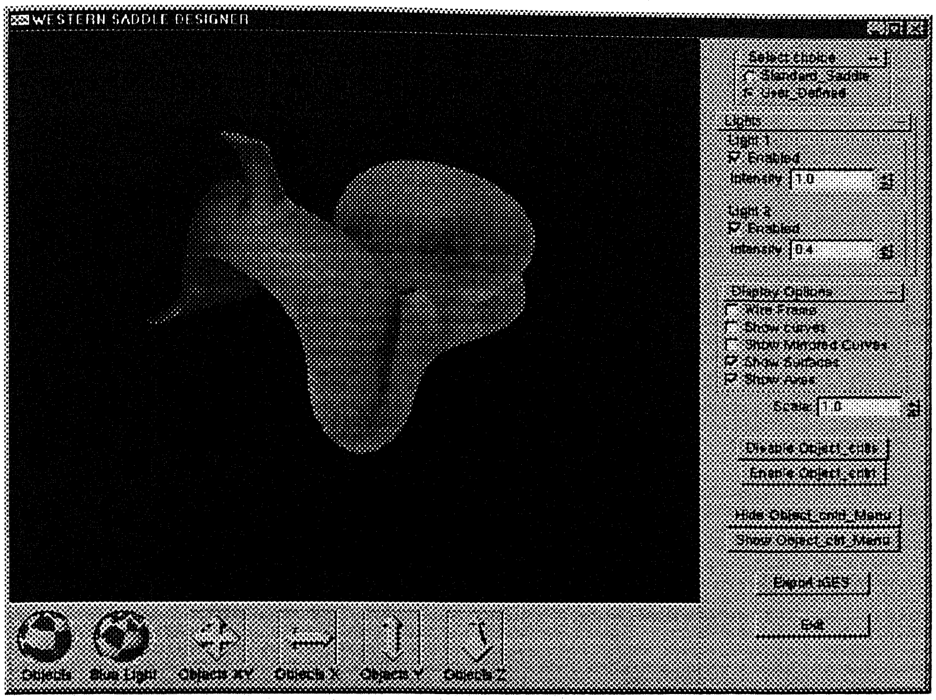


Fig 4.7 User Customization Example1 (surface model)

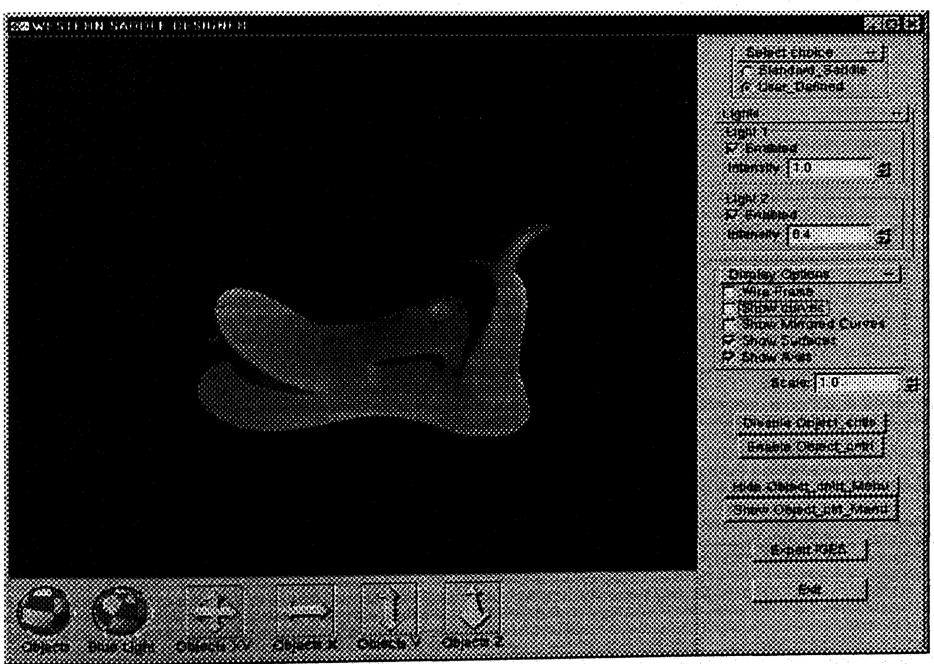


Fig 4.8 User Customization Example2 (surface model)

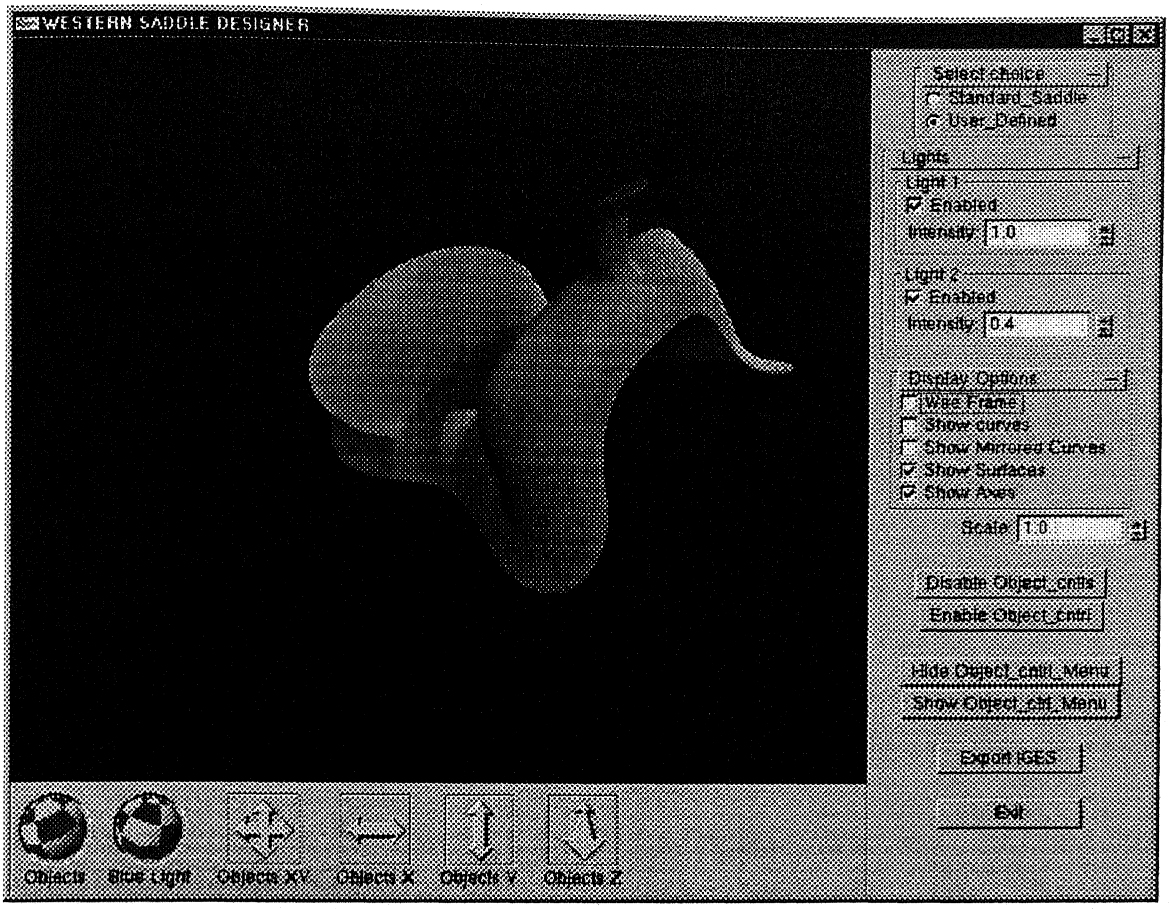


Fig 4.9 User Customization Example2 (surface model)

4.5 Data Transfer

The surface model developed using the software has to be transferred to the commercial CAD/CAM software for further processing which may include design modification and for down stream applications like simulation, prototyping and CNC machining etc. Universal data formats like IGES, VDA, STEP etc can be used for the data transfer. The software implementation exports the design data in IGES file format.

4.5.1 CAD/CAM Data Exchange

One of the big obstacles in CAD/CAM world used to be the diversity of the data formats that were used by the different CAD/CAM systems. It becomes increasingly

important to find effective procedures for exchanging these data among CAD/CAM systems.

Transferring data between dissimilar CAD/CAM systems must embrace the complete product description stored in its database. The complete product description is consists of four types of modeling data, these are shape, nonshape, design, and manufacturing data. Shape data consists of both geometrical and topological information. Nonshape data includes graphics data such as shaded images, and model global data as measuring units of database and resolution of storing the data base numerical values. Design data is the information that designers generate from geometric model for the analysis purpose. Manufacturing data consists of information as tooling, NC tool paths, tolerance, process planning etc.

IGES data format is focused on CAD-to-CAD exchange where primarily shape and nonshape data are transferred from one system to another. On the contrary PDES is an example of data format, which transfers all four types of modeling data discussed earlier.

4.5.2 Data Translator Types

Data translators are mainly classified in to two types,

1.Direct translator.

2.Indirect translator.

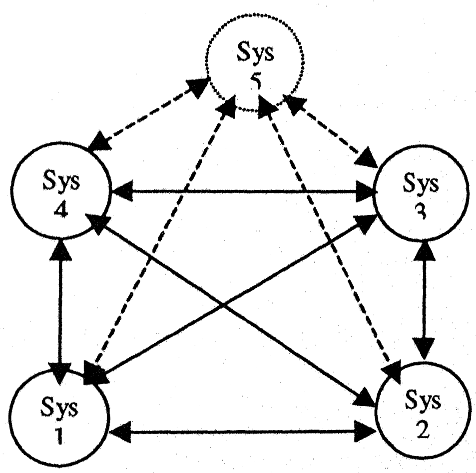


Fig 4.10a. Direct Translator

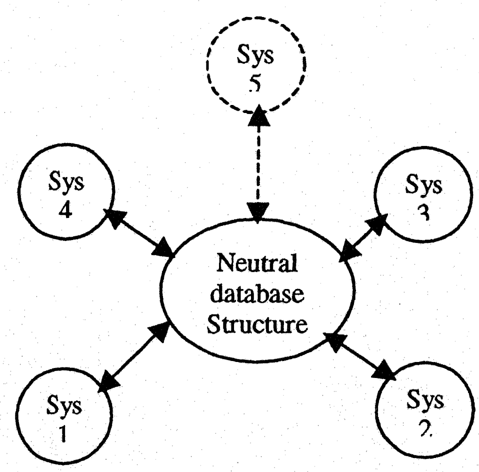


Fig.4.10b Indirect Translator

4.5.2.1 Direct Translator

The direct translator translates the model data stored in the product database directly from one CAD/CAM system format to another in one step. Fig 4.10a shows the schematic diagram of the direct translator. These are typically written by the computer service companies that are specialized in CAD/CAM database conversion. Direct translators are considered to be a dedicated translator programs.

4.5.2.2 Indirect Translator

Indirect translator is more general and adopts the philosophy of creating a neutral data base structure, which is independent of existing or future CAD/CAM systems. Fig 4.10b shows the functioning of the indirect translator. Each CAD/CAM system has its own processor to transfer data to and from neutral data format.

4.5.3. IGES Data Format

IGES is the first standard exchange format developed to address the concept of communicating product data among dissimilar CAD/CAM systems. IGES is viewed as a communication file for transmitting data between two systems or applications. IGES defines a neutral database in the form of file format, which describes an "IGES model" of model data of a given product. The IGES model can be read and interpreted by dissimilar CAD/CAM systems. Fig 4.11 shows the data base exchange using IGES. The software that translates from the native database format of a given CAD/CAM system to the IGES format is called preprocessor. The software that translates in opposite way (from IGES to CAD/CAM system) is called a postprocessor.

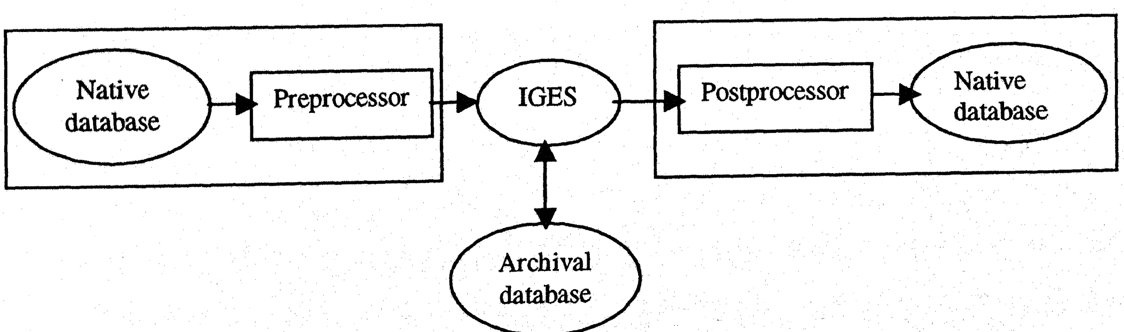


Fig4.11 Database Exchange using IGES

4.5.3.1. General structure of IGES format

An IGES file is a sequential file consists of sequence of records. The file format treats the product definitions to be exchanged as a file of entities, each entity being represented in a standard format, to and from which the native representation of a specific CAD/CAM system can be mapped. Depending on the chosen format, the record length can be fixed or variable. There are two different formats to represent IGES data in a file: ASCII and binary. The ASCII has two format types: a fixed 80-character record length format and a compressed format. In fixed record length format the entire file is of 80-character length. The IGES data written in columns 1 through 72 inclusive of each record, column 73 stores the section identification character, and column 74 to 80 are specified for the section sequence number of each record.

IGES file consists of six sections in the fallowing order as shown in fig 4.12

1.Flag section (optional)

2.Start section.

3.Global section

4.Directory entry section (DE).

5.Parameter data (PD) section, and

6.Terminate section.

The identification character, also called the section code for these sections respectively are S, G, D, P, and T (excluding flag section). The flag section is only used with the compressed data format, which is not using in the present work. It is single records line that precedes the start section with a character "C" in column 73 to identify the file as compressed ASCII. Start section is a human readable introduction to the file. It is commonly described as a "prologue" to IGES file. The global section includes information describing the processor and information needed by the postprocessor to interpret the file. Some of the parameters that are: characters used as delimiters between individual entries and between records, the name of IGES file itself, vendor and software version of sending system, model space scale model units, minimum resolution and maximum coordinate values etc. The DE section is the list of all the entities defined in the IGES file together with certain attributes associated with them. The entry for each

entity occupies two 80-character records which are devided into a total of twenty 8-character fields. This section consists of entity type number, pointer to the parameter data entry in PD section, sequence number in DE section itself, line font, layer number, transformation matrix, line weight, and color etc. the PD section consists of an actual data defining the each entity listed in the DE section like line, surface etc. parameter data is placed in free format in columns 1 through 64. The parameter delimiter is used to separate parameters and record delimiter is used to terminate the list of parameters. Both these delimiters are specified in the global section of IGES file. Column 65 is left blank. Column 66 to 72 on all PD records contains the entity pointer to the first record of the entity in the DE section. The terminate section contains a single record which specifies the number of each records in each of the four preceding sections for checking purposes.

In the compressed ASCII format, the start, global, and terminate section remains same as ASCII form, while the DE and PD sections are combined in to a single data section. In addition, each record of PD portion of data section is written in free form similar to the ASCII PD section, but is of variable length to eliminate storing space character.

The binary format is a bit stream binary representation of data. All entry parameterization and data organization are otherwise identical to the ASCII form. Each section in the fiie begins with the section code, that is, S, G, D, P, or T, fallowed by the section count to specify the total bites that belong to the section. DE and PD sections remain same as the ASCII format.

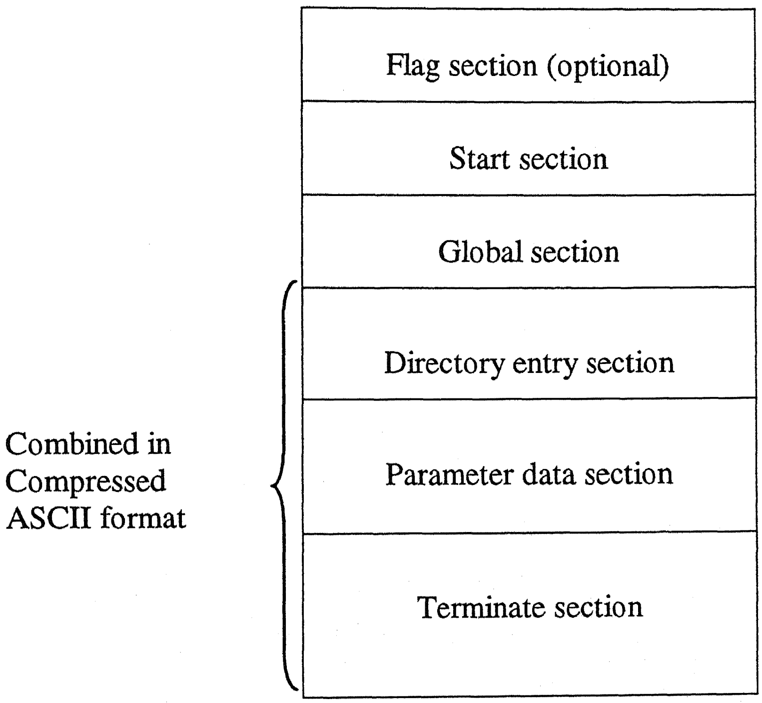


Fig 4.12 IGES file general structure

4.5.4 Representation of NURBS in IGES

In the present work we are exporting only B-spline surfaces from native database. The Tables 4.1 and 4.2 show how the B-spline curves and surfaces can be represented in IGES format.

Name	Type	Description
N	Int	number of control points (counted from 0)
n	int	degree
flag1	int	0:non planar, 1:planar
flag2	int	0:open, 1: closed
flag3	int	0:rational, 1: integral
flag4	int	0: non periodic, 1:periodic
		Knot sequence
u_{-n}	double	first knot
:	:	:
u_{N+1}	double	last knot
		Weights
w_0	double	first weight
:	:	:
w_N	double	last weight
		Control points
d_0^x	double	x-component of first control points
d_0^y	double	y-component of first control points
d_0^z	double	z-component of first control points
:	:	:
d_N^x	double	x-component of last control points
d_N^y	double	y-component of last control points
d_N^z	double	z-component of last control points
		Range
u_{start}	double	Starting parameter value
u_{end}	double	Ending parameter value
		Plane normal (if flag1 =1)
n^x	double	x-component of normal
n^y	double	y-component of normal
n^z	double	z-component of normal
		IGES format 126:NURBS curves

Table 4.1 Representation of NURBS curve in IGES format

Name	Type	Description
M	Int	number of control points (counted from 0)in u-direction
N	int	number of control points (counted from 0)in v-direction
m	int	degree in u-direction
n	int	degree in v-direction
flag1	int	1:closed in u-direction, 0:else
flag2	int	1:closed in v-direction, 0:else
flag3	int	0:rational, 1: integral
flag4	int	0: non periodic u-direction, 1:periodic
flag5	int	0: non periodic v-direction, 1:periodic
	Knot sequence	
u_{-m}	Double	first knot in u-direction
:	:	:
u_{M+1}	double	last knot in u-direction
v_{-n}	Double	first knot in v-direction
:	:	:
v_{N+1}	double	last knot in u direction
	Weights	
$w_{0,0}$	Double	first weight
$w_{1,0}$	double	second weight
:	:	:
$w_{M,N}$	double	last weight
	Control points	
$d^x_{0,0}$	double	x-component of first control points
$d^y_{0,0}$	double	y-component of first control points
$d^z_{0,0}$	double	z-component of first control points
$d^x_{1,0}$	double	x-component of second control points
$d^y_{1,0}$	double	y-component of second control points
$d^z_{1,0}$	double	z-component of second control points
:	:	:
$d^x_{M,N}$	double	x-component of last control points
$d^y_{M,N}$	double	y-component of last control points
$d^z_{M,N}$	double	z-component of last control points
	Range	
u_{start}	Double	Starting parameter value in u-direction
u_{end}	double	Ending parameter value in u-direction
v_{start}	double	Starting parameter value in v-direction
v_{end}	double	Ending parameter value in v-direction
	IGES format 126:NURBS curves	

Table 4.2 Representation of NURBS surface in IGES format

Chapter 5
Conclusions

The present work highlights the semantic featured based modeling of a free-form sculptured object. The free-form object taken for the present study is western saddle tree, which forms the core of the saddle.

5.1 Technical Summary

Modeling a sculptured object, like saddle tree, by the proposed feature language is elaborated as follows:

(i) defining the required object and feature vocabularies
(ii) defining the grammatical rules and
(iii) manipulating the vocabularies according to the grammar to describe the object.

The modeling approach conforms to the approach of aggregating a set of features to model a regular shaped object, except for the vocabulary and grammar definitions. The feature language provides a generic mechanism for defining and manipulating features. It gives the user an option of building a custom feature library that might satisfy a specific application. For a regular shaped object, the feature semantics are represented by a set of analytical geometry, while the rules manipulating the feature relationships are expressed in terms of the boolean operations. The definitions are geometry oriented. Table 5.1 lists the similarities of the traditional and proposed feature modeling for regular shaped and sculptured object, respectively. There are two main differences between a feature for regular shaped object and sculptured object:

(i) *The dependence of the feature semantics upon the connectivity of a feature within the object configuration:* For regular shaped object, the feature semantic is independent on the feature connectivity with its neighbouring features within the feature model.

For example, a hole remains a hole whatever its position in the object. On the contrary, the semantics of a feature for a sculptured object are destroyed if the feature connectivity is changed within the model.

(ii) *The arbitrary free-form geometry of a sculptured objects:* The geometry of a regular shaped object is restricted to analytical geometry, which leads to finite variation. The arbitrary free-form geometry of a sculptured object gives infinite variation of a shape. As a result, the geometry variation of the sculptured object class is higher than that of the regular object class.

Applying the geometry oriented approach to define object and feature vocabularies for sculptured objects results in a specific feature category for either free-form surfaces or each single free-form surface [7-9] due to the infinite form variation of free-form surfaces. Both cases have their drawbacks. The former will be to general and difficult to identify from each other, while the latter will cause a very large feature taxonomy. In order to overcome these obstacles, a constituent features set and a neighbouring feature set are included in the contents of the vocabulary object and feature.

Geometric Components	Traditional feature modeling for regular shaped object	Proposed feature modeling for sculptured object
Features	Pre-defined in the feature library, i.e. holes, slot, etc.	User-defined vocabularies and object specific
Feature Operating Rules	Boolean operation	User-defined Grammatical rules

Table 5.1 The Similarities of the Traditional and Proposed Feature Modeling Approach

Furthermore, the boolean operations do not consider the surface continuity of the operands, while the proposed edge vocabularies emphasize this geometric characteristic.

For example, a hole remains a hole whatever its position in the object. On the contrary, the semantics of a feature for a sculptured object are destroyed if the feature connectivity is changed within the model.

(ii) *The arbitrary free-form geometry of a sculptured objects:* The geometry of a regular shaped object is restricted to analytical geometry, which leads to finite variation. The arbitrary free-form geometry of a sculptured object gives infinite variation of a shape. As a result, the geometry variation of the sculptured object class is higher than that of the regular object class.

Applying the geometry oriented approach to define object and feature vocabularies for sculptured objects results in a specific feature category for either free-form surfaces or each single free-form surface [7-9] due to the infinite form variation of free-form surfaces. Both cases have their drawbacks. The former will be to general and difficult to identify from each other, while the latter will cause a very large feature taxonomy. In order to overcome these obstacles, a constituent features set and a neighbouring feature set are included in the contents of the vocabulary object and feature.

Geometric Components	Traditional feature modeling for regular shaped object	Proposed feature modeling for sculptured object
Features	Pre-defined in the feature library, i.e. holes, slot, etc.	User-defined vocabularies and object specific
Feature Operating Rules	Boolean operation	User-defined Grammatical rules

Table 5.1 The Similarities of the Traditional and Proposed Feature Modeling Approach

Furthermore, the boolean operations do not consider the surface continuity of the operands, while the proposed edge vocabularies emphasize this geometric characteristic.

Fig.5.1 shows the feature modeling for regular shaped objects and the proposed modeling approach for sculptured objects. For the regular shaped object class **R**, there is a common feature set as $\mathbf{F_R}$ such as holes, slots etc. for modeling the elements ($\mathbf{r}_1$, ..., $\mathbf{r}_k$) of **R**. However, there is no such set of features pertinent to sculptured objects. Each element of the sculptured object class $\mathbf{S}_i$ (where $\mathbf{S} = \cup_i \mathbf{S}_i$) has a specific feature class $\mathbf{F}_{Si}$ as the vocabularies of the feature language.

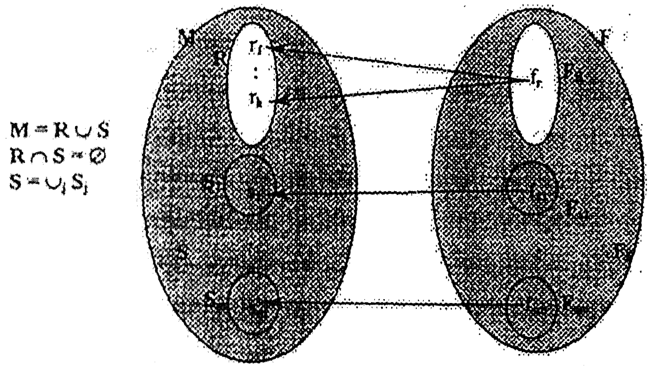


Fig.5.1 The feature modeling of regular shaped objects and sculptured objects

The geometric description of a sculptured object is basically defined by a set of surfaces in the feature vocabulary and continuity constraints from the feature-feature link. The complexity of the problem depends upon the surface representation and the order of continuity.

5.2 Concluding Remarks

An initial feature anatomy for a western type of saddletree is presented, to support the need for establishing a generic approach of saddletree design. The following remarks can be made based on this work.

(1) It is possible to decompose a saddle tree into feature anatomy for proposed "semantic" feature-based design.

(2) There are issues common to sculptured products, including saddle trees, that are specifically addressed and dealt effectively within the proposed "semantic" feature-based design method, such as the following:

- CAD tool implementation based on product-specific terminology;
- localized and simplified shape control for sculptured product design;
- hybrid and variational product design;
- anisotropic grading of product families.

(3) Proposed "semantic" feature-based design approach shows significant potential to benefit saddle tree design efficiency, enhance size grading and improve manufacturing processes.

5.3. Recommendations for Future Work

- In the present work, a simple method for solving the continuity problems is used. One can use a better, robust method for solving the continuity problems.
- Uniform cubic B-spline curves are used for modeling the boundary curves. NURBS can be used for better results.
- The proposed generic model is for western type of saddle tree. It can be extended to generic feature model which can represent all types of saddle trees.
- Once the generic feature model is developed, some procedures have to be developed for using this generic feature model in reverse engineering applications wherein a user will input the digitized data for any available saddle tree and can get the surface model for the same.
- The present software exports data in IGES format. Algorithms can be developed for exporting in other formats such as STL, VDA etc.

References

[1] Choi B.K., (1991), 'Surface Modeling for CAD/CAM', Elsevier Science Publishers B.V., The Netherlands.

[2] Mortenson, M.E. (1985), 'Geometric Modeling', John Wiley & Sons, New York.

[3] Mitchell, S.R., Jones, R. and Newman S.T., (1995), 'A structured approach to the design of shoe lasts', *Journal of Engineering Design*, Vol. 6, No. 2, pp. 149-166.

[4] Au, C.K. and Yuen, M.M.F., (1999), 'Feature-based reverse engineering of mannequin for garment design', *Computer-Aided Design*, Vol. 31, No. 12, pp. 751-759.

[5] Flasinski, M., (1995), 'Use of graph grammars for the description of mechanical parts', *Computer-Aided Design*, Vol. 27, No. 6, pp. 403-433.

[6] Longenecker, S.N. and Fitzhorn, P.A., (1988), 'Form + Function + algebra = feature grammars', In: Newsome SL, Spillers, W.R., Finger, S., editors, Design Theory'88, New York: Springer, pp.189-207.

[7] Mitchell, S.R., Jones, R. and Hinde, Chris, (1995), 'An initial data model, using the object-oriented paradigm, for sculptured feature-based design', *Research in Engineering Design*, Vol.7, pp. 19-37.

[8] Cavendish, J.C., (1995), 'Integrating feature-based surface design with freeform deformation', *Computer-Aided Design*, Vol. 27, No. 9, pp. 703-711.

[9] Van Elsas, P.A. and Vergeest J.S.M., (1998), 'Displacement feature modelling for conceptual design', *Computer-Aided Design*, Vol. 30, No. 1, pp. 19-27.

[10] Kim, C. and O'Grady, R., (1996), 'A representation formalism for feature-based design', *Computer-Aided Design*, Vol. 28, No. 6/7, pp. 451-460.

[11] Hoffmann, C.M. and Joan-Arinyo, R., (1998), 'On user-defined features', *Computer-Aided Design*, Vol. 30, No. 5, pp. 321-332.

[12] Au, C.K. and Yuen, M.M.F., (2000), 'A semantic feature language for sculptured object modelling', *Computer-Aided Design*, Vol. 32, No. 1, pp. 63-74.

[13] Waykole, V. S. (2001) 'Feature-based design of saddle tree shapes', *M.Tech Thesis*, Department of mechanical Engineering, I.I.T. Kanpur.

[14] Bidarra, R. and Broonsvoort, W.F. (2000), 'Semantic feature modelling', *Computer-Aided Design*, Vol. 32, No. 3, pp. 201-225.

[15] Ibrahim Zied (1998), *CAD/CAM Theory and Practice*, Tata McGraw-Hill Edition.

Calculation of Characteristic Points

As stated in Section 3.2.2.2, all the characteristic points are calculated as some proportion of the global sizing parameters namely, length (*l*), depth (*d*) and width (*w*). These proportions are presented in table A1.1.

Every curve is represented by 4 control points. The endpoints of every curve are the nothing but the characteristic points.

Let, 1_A, 1_B, 1_C and 1_D represent the four control points for curve no. 1. Then 1_A and 1_D will represent the endpoints/characteristic points for curve no. 1. All such characteristic points are shown in Fig. A1.1 and Fig. A1.2.

Characteristic point	**X Co-ordinate**	**Y Co-ordinate**	**Z Co-ordinate**
0_A, 1_A	$0.0\,l$	$0.0\,w$	0.0
1_D, 2_A, 72_A	$0.136\,l$	$0.5515\,w$	$-0.378\,d$
0_D, 3_A, 4_A	$0_{AX}+0.2973l$	1_{AY}	$-1.0\,d$
2_D, 3_D, 5_A, 27_A	0_{DX}	$0_{AY}+0.4595\,w$	$0_{AZ}-0.23d$
4_D, 6_A, 7_A	$1_{AX}+0.58\,l$	$1_{AY}-0.4\,w$	0.0
7_D, 9_A, 10_A	$4_{AX}+0.5557\,l$	1_{AY}	1_{AZ}
5_D, 6_D, 8_A, 28_A	7_{AX}	$7_{AY}+0.298\,w$	$-0.904\,d$
8_D, 9_D, 11A, 37_A	9_{AX}	$9_{AY}+0.835\,w$	$-0.376\,d$
10_D, 12_A, 133_A	$1.0\,l$	9_{AY}	$-0.338\,d$
133_D, 12_D, 14_A	$4_{AX}+0.637\,l$	$0.5\,w$	$1.032\,d$
54_A, 53_D, 100_D	12_{DX}	12_{DY}	12_{DZ}
133_D, 13_A, 134_A, 16_A, 17_D	$12_{AX}+0.058\,l$	12_{DY}	$1.3\,d$
13_D, 15_A	$12_{AX}+0.25\,l$	12_{DY}	$0.636\,d$
17_A, 19_A, 20_D	$1.09\,l$	1_{AY}	$0_{DZ}+0.604\,d$

Calculation of Characteristic Points

As stated in Section 3.2.2.2, all the characteristic points are calculated as some proportion of the global sizing parameters namely, length (l), depth (d) and width (w). These proportions are presented in table A1.1.

Every curve is represented by 4 control points. The endpoints of every curve are the nothing but the characteristic points.

Let, 1_A, 1_B, 1_C and 1_D represent the four control points for curve no. 1. Then 1_A and 1_D will represent the endpoints/characteristic points for curve no. 1. All such characteristic points are shown in Fig. A1.1 and Fig. A1.2.

Characteristic point	**X Co-ordinate**	**Y Co-ordinate**	**Z Co-ordinate**
0_A, 1_A	$0.0\,l$	$0.0\,w$	0.0
1_D, 2_A, 72_A	$0.136\,l$	$0.5515\,w$	$-0.378\,d$
0_D, 3_A, 4_A	$0_{AX}+0.2973l$	1_{AY}	$-1.0\,d$
2_D, 3_D, 5_A, 27_A	0_{DX}	$0_{AY}+0.4595\,w$	$0_{AZ}-0.23d$
4_D, 6_A, 7_A	$1_{AX}+0.58\,l$	$1_{AY}-0.4\,w$	0.0
7_D, 9_A, 10_A	$4_{AX}+0.5557\,l$	1_{AY}	1_{AZ}
5_D, 6_D, 8_A, 28_A	7_{AX}	$7_{AY}+0.298\,w$	$-0.904\,d$
8_D, 9_D, 11A, 37_A	9_{AX}	$9_{AY}+0.835\,w$	$-0.376\,d$
10_D, 12_A, 133_A	$1.0\,l$	9_{AY}	$-0.338\,d$
133_D, 12_D, 14_A	$4_{AX}+0.637\,l$	$0.5\,w$	$1.032\,d$
54_A, 53_D, 100_D	12_{DX}	12_{DY}	12_{DZ}
133_D, 13_A, 134_A, 16_A, 17_D	$12_{AX}+0.058\,l$	12_{DY}	$1.3\,d$
13_D, 15_A	$12_{AX}+0.25\,l$	12_{DY}	$0.636\,d$
17_A, 19_A, 20_D	$1.09\,l$	1_{AY}	$0_{DZ}+0.604\,d$

$20_A, 22_A, 23_A$	$12_{AX}+ 0.15\ l$	0.0	$0_{DZ}+1.3\ d$
$23_D, 25_A, 26_A$	$12_{AX}+ 0.18\ l$	4_{DY}	$2.44\ d$
$25_D, 26_D, 24_D$	$25A+0.133\ l$	0.0	$25_A +0.267\ d$
$24_D, 22_D, 21_D$	$12_{AX}+ 0.22\ l$	0.0	$2.18\ d$
$21_A, 19_D, 18_D$	$12_{AX}+ 0.19\ l$	0.0	$1.46\ d$
$15_D, 14_D, 57_D$	$12_{AX}+ 0.2\ l$	$0.344\ w$	$0.554\ d$
$57_A, 56_D$	$12_{AX}+ 0.233\ l$	$14_{DY}+0.184\ w$	$0_{DZ}-0.46\ d$
$54_D, 55_D, 56_A$	14_{DX}	$0_{DY} + 0.526\ w$	$0_{DZ} - 0.46\ d$
$55_A, 46_D, 45_D$	9_{DX}	$0_{DY} + 0.59\ w$	$0_{DZ} + 0.034\ d$
$52_A, 51_A, 38_D, 48_D, 49_D$	$12_{AX}-0.11\ l$	$0.407\ w$	$0_{DZ} + 0.324\ d$
$49_D, 50_A, 47_D, 101_A$	$12_{AX}-0.156\ l$	$0_{DY} + 0.194\ w$	$0.886\ d$
$47_A, 42_D, 41_D, 110_D$	9_{DX}	$0.182\ w$	$-0.82\ d$
$42_A, 40_D$	9_{DX}	$0_{DY} + 0.342\ w$	$-0.48\ d$
$41_A, 39_D, 35_A, 117_D$	4_{DX}	$0_{DY} + 0.158\ w$	$0_{DZ} - 0.152\ d$
$40_A, 39_A, 34_A$	41_{AX}	$0_{DY} + 0.284\ w$	$0.837\ d$
$34_D, 32_D$, 30A	91_{AX}	91_{AY}	91_{AZ}
$35_D, 32_A, 33_A, 121_A$	119_{DX}	119_{DY}	119_{DZ}
$45_A, 43_D, 31_D$	4_{DX}	$0.59\ w$	$0_{DZ} - 0.91\ d$
$31_A, 28_D, 29_D$	4_{DX}	$0_{DY} + 0.521\ w$	$0_{DZ} - 0.87\ d$
$27_D, 29_A, 70_D, 82_D$	0_{AX}	$0.588\ w$	$0_{DZ} - 0.898\ d$
$69_D, 82_A$	81_{DX}	81_{DY}	81_{DZ}
$63_D, 62_D$	80_{DX}	80_{DY}	80_{DZ}
$58_D, 61_A, 62_A$	$0.13\ l$	$0.559\ w$	$-0.412\ d$
$63_A, 61_D, 59_D$	$0_{AX} +0.163l$	$0_{DY} + 0.417\ w$	$0_{DZ} - 0.26\ d$
$58_A, 60_A$	$0_{AX} - 0.01l$	0.0	$0.033\ d$
$59_A, 60_D$	$0_{AX} - 0.05l$	0.0	$0_{DZ} - 0.09\ d$
$69_A, 68_A, 65_D$	$0.156\ l$	$0.424\ w$	$-1.38\ d$

68_D, 70A, 67_D	0_{AX} - 0.075l	0.581 w	0_{DZ} - 0.916 d
104_A, 102_A	12_{AX}-0.242 l	0.0	0_{DZ} - 0.606 d
101_A, 102_D, 131_A	1.194 l	12_{AY}+0.33 w	-0.54 d
104_D, 103_D, 105_D, 106_D	79_{AX}	0.0	0.02 d
107_D, 109_A, 111_A, 108_D	78_{AX}	0_{AX}	-0.207 d
111_D, 110_A, 118_A	47_{AX}	0.0	0.2656 d
131_D, 132_A	12_{AX}+0.23 w	12_{AY}+0.527 w	-1.566 d
132_D, 112_D, 114_A	78_{AX}	12_{AY}+0.586 w	-1.951 d
113_D, 114_D, 116_D	113_{AX}	110_{AY}+0.579 w	0_{DX}+ 0.949 d
116_A, 115_D, 122_A, 126_D	43_{AX}	0.506 w	0_{DX}+ 0.915 d
117_A, 118_D, 120_A	39_{AX}	0.0	0_{DX}+ 0.07 d
120_D, 125_A, 123_A, 119_A	0.64 l	0.0	1.1 d
123_D, 126_A	123_{AX}	0.506 w	-1.916 d
125_D, 124_A	119_{AY}	0_{DY}	119_{AZ}
124_D, 126_D, 130_A, 128_D	119_{AY}	12_{AY}+0.574 w	2(0_{AZ})
129_D, 130_D	0_{AX}+ 0.0446 l	12_{AY}+0.359 w	0_{DX}+ 0.68 d

Table A1.1 Characteristic Points and Their Values

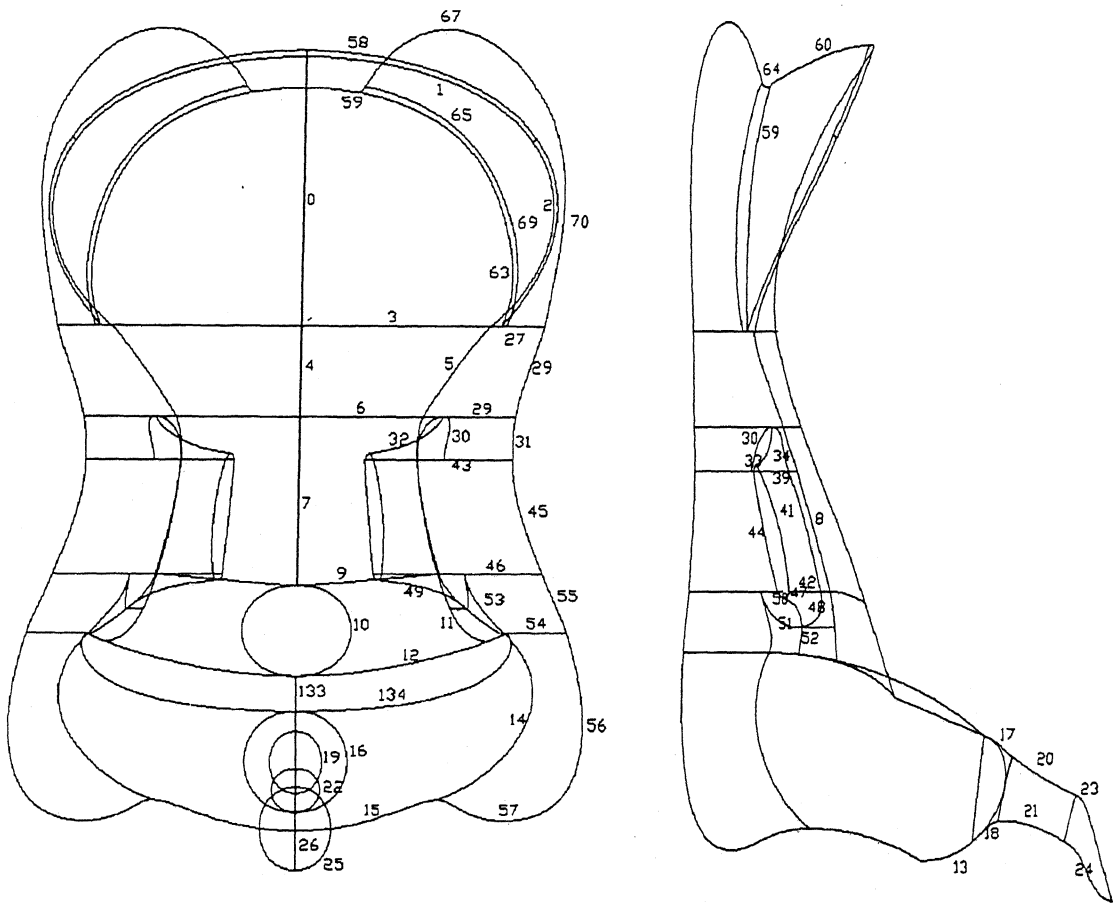


Fig. A1.1 All Curves And Characteristic Points (Upper Side)

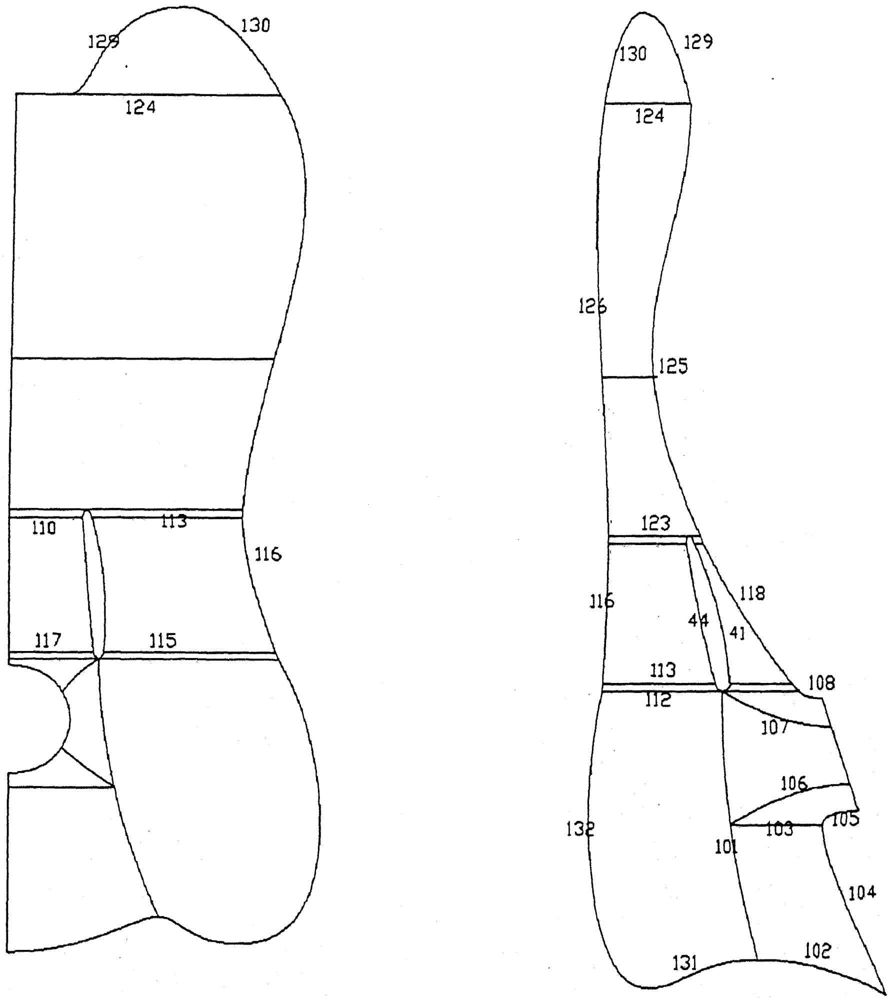


Fig. A1.2 All Curves And Characteristic Points (Lower Side)

Calculation of Intermediate Control Points

A2.1 Curves – Upper and Lower Side

The intermediate control points for all curves are calculated using eq. 3.1. The values of k and v are the key factors in eq. 3.1. These k and v values vary for every curve, every intermediate control point. All these values are presented through Table A2.1 below.

Curve No.	Control Point	k	Direction Vector, v		
			v_x	v_y	v_z
0	0_B	0.332	Cos (α)*	0.0	Sin (α)
	0_C	0.332	0.991	0.0	0.136
1	1_B	0.333	0.0	1.0	0.0
	1_C	0.333	0.76	0.522	-0.387
2	2_B	0.372	0.76	0.522	-0.387
	2_C	0.372	0.66	-0.744	-0.10
3	3_B	0.325	0.0	1.0	0.0
	3_C	0.325	0.0	0.82	-0.572
4	4_B	0.181	0.991	0.0	0.136
	4_C	0.181	0.920	0.0	0.393
5	5_B	0.178	0.66	-0.744	-0.10
	5_C	0.178	0.864	-0.424	0.324
6	6_B	0.187	0.0	1.0	0.0
	6_C	0.187	0.0	0.883	-0.469
7	7_B	0.3413	0.920	0.0	0.393
	7_C	0.3413	Cos (β)**	0.0	Sin (β)

8	8_B	0.211	0.864	-0.424	0.324
	8_C	0.211	0.924	0.292	0.247
9	9_B	0.189	0.0	1.0	0.0
	9_C	0.189	0.0	0.643	-0.766
10	10_B	0.725	0.00	1.00	0.00
	10_C.	0.725	0.00	-1.00	0.00
11	11_B	0.350	0.924	0.292	0.247
	11_C	0.350	0.10	0.99	0.10
12	12_B	0.4433	0.00	1.00	0.00
	12_C	0.4433	0.00	-0.10	-0.995
13	13_B	0.414	$T_{X133}(1.0)$#	$T_{Y133}(1.0)$	$T_{Z133}(1.0)$
	13_C	0.618	0.00	0.00	-1.00
14	14_B	0.564	0.644	0.656	-0.36
	14_C	0.42	0.301	-0.803	0.515
15	15_B	0.404	0.00	1.00	0.00
	15_C	0.404	0.00	-Sin (γ)	Cos (γ)
16	16_B	0.658	0.00	1.00	0.00
	16_C	0.658	0.00	-1.00	0.00
17	17_B	0.502	$T_{X133}(1.0)$	$T_{Y133}(1.0)$	$T_{Z133}(1.0)$
	17_C	0.502	0.564	0.00	0.826
18	18_B	0.29	$T_{X13}(0.22)$	$T_{X13}(0.220)$	$T_{X13}(0.220)$
	18_C	0.29	-0.113	0.00	0.994
19	19_B	0.526	0.00	1.00	0.00
	19_C	0.526	0.00	-1.00	0.00
20	20_B	0.336	0.564	0.00	0.826
	20_C	0.336	0.367	0.00	0.930
21	21_B	0.340	-0.113	0.00	0.994
	21_C	0.340	0.439	0.00	0.899
22	22_B	0.693	0.00	1.00	0.00

	22_C	0.693	0.00	-1.00	0.00
23	23_B	0.350	0.367	0.00	0.930
	23_C	0.350	0.949	0.00	0.315
24	24_B	0.416	0.439	0.00	0.899
	24_C	0.416	0.293	0.00	0.956
25	25_B	0.535	0.00	1.00	0.00
	25_C	0.535	0.00	-1.00	0.00
26	26_B	0.333	0.949	0.00	0.315
	26_C	0.333	0.949	0.00	0.315
27	27_B	0.306	0.00	0.82	-0.572
	27_C	0.306	0.00	0.485	-0.875
28	28_B	0.321	0.00	0.883	-0.469
	28_C	0.321	0.00	0.469	-0.883
29	29_B	0.354	0.963	-0.263	0.052
	29_C	0.354	0.983	-0.179	0.0453
30	30_B	0.343	$T_{X91}(1.0)$	$T_{Y91}(1.0)$	$T_{Z91}(1.0)$
	30_C	0.343	1.00	0.00	0.00
31	31_B	0.335	0.983	-0.179	0.0453
	31_C	0.335	1.00	0.00	0.00
32	32_B	0.340	-0.126	0.950	0.286
	32_C	0.340	-0.86	0.509	0.003
33	33_B	0.336	0.00	0.707	-0.707
	33_C	0.336	0.951	0.254	0.179
34	34_B	0.405	-0.919	-0.089	-0.385
	34_C	0.405	$T_{X91}(1.0)$	$T_{Y91}(1.0)$	$T_{Z91}(1.0)$
35	35_B	0.368	-0.884	-0.0612	-0.463
	35_C	0.368	0.00	0.707	-0.707
39	39_B	0.343	0.000	0.965	0.262

	39_C	0.343	0.00	0.644	0.765
40	40_B	0.334	0.919	0.089	0.385
	40_C	0.334	0.933	0.284	0.222
41	41_B	0.335	0.884	0.0612	0.463
	41_C	0.335	0.991	0.097	0.092
42	42_B	0.318	0.00	0.98	0.20
	42_C	0.318	0.00	0.705	0.709
43	43_B	0.392	0.00	0.914	0.405
	43_C	0.392	0.00	0.547	-0.837
44	44_B	0.335	0.951	0.254	0.179
	44_C	0.335	0.949	-0.097	0.299
45	45_B	0.336	1.00	0.000	0.00
	45_C	0.336	0.933	0.355	-0.056
46	46_B	0.321	0.00	0.876	0.470
	46_C	0.321	0.00	0.601	-0.798
47	47_B	0.383	0.991	0.097	0.092
	47_C	0.383	0.00	0.533	-0.846
48	48_B	0.414	0.933	0.284	0.222
	48_C	0.414	0.163	0.19	-0.968
49	49_B	0.338	0.121	0.955	0.268
	49_C	0.338	0.67	0.739	0.072
50	50_B	0.332	0.00	0.533	-0.846
	50_C	0.332	-0.949	0.097	-0.299
51	51_B	0.376	0.163	0.19	-0.968
	51_C	0.376	-0.942	-0.162	-0.293
52	52_B	0.280	0.670	0.739	0.072
	52_C	0.280	0.38	0.915	0.136
53	53_B	0.33	0.948	0.153	0.278
	53_C	0.33	0.664	0.656	-0.36

54	54_B	0.371	0.00	0.10	0.995
	54_C	0.371	0.00	0.644	-0.765
55	55_B	0.325	0.933	0.355	-0.056
	55_C	0.325	0.950	0.309	-0.044
56	56_B	0.453	0.950	0.309	-0.044
	56_C	0.453	0.056	-0.835	0.547
57	57_B	0.453	0.056	-0.835	0.547
	57_C	0.453	0.00	0.312	0.950
58	58_B	0.333	0.00	1.00	0.00
	58_C	0.333	0.76	0.522	-0.387
59	59_B	0.345	0.00	1.00	0.00
	59_C	0.345	0.794	0.593	-0.131
60	60_B	0.338	0.055	0.00	-00988
	60_C	0.338	0.444	0.00	-0.896
61	61_B	0.335	0.064	0.518	0.853
	61_C	0.335	0.250	-0.5589	-0.790
62	62_B	0.377	0.76	0.552	-0.387
	62_C	0.377	0.742	-0.627	-0.236
63	63_B	0.305	0.794	0.593	-0.131
	63_C	0.305	0.906	-0.420	0.044
65	65_B	0.343	0.184	0.977	-0.107
	65_C	0.343	0.791	0.591	-0.159
67	67_B	0.535	-0.838	0.474	-0.270
	67_C	0.535	0.872	0.471	-0.133
68	68_B	0.336	-0.532	0.692	-0.487
	68_C	0.336	-0.494	0.598	-0.631
69	69_B	0.336	0.791	0.591	-0.159
	69_C	0.336	0.916	-0.343	0.208
70	70_B	0.366	0.872	0.471	-0.133

	70_C	0.366	0.983	-0.179	0.0453
72	72_B	0.588	-0.566	0.819	0.093
	72_C	0.588	0.064	-0.518	-0.853
102	102_B	0.404	0.00	1.00	0.00
	102_C	0.404	Cos (γ)***	0.0	Sin (γ)
103	103_B	0.400	0.00	1.00	0.00
	103_C	0.400	0.00	0.408	-0.913
104	104_B	0.21	1.00	0.00	0.00
	104_C	0.21	0.909	0.00	0.417
105	105_B	0.278	0.00	0.00	-1.00
	105_C	0.278	1.00	0.00	0.00
106	106_B	0.331	0.909	0.00	0.417
	106_C	0.331	0.00	0.00	-1.00
107	107_B	0.331	0.909	0.00	0.417
	107_C	0.331	0.00	0.00	-1.00
108	108_B	0.325	0.00	0.00	-1.00
	108_C	0.325	-0.707	0.00	-0.707
109	109_B	0.419	0.00	1.00	0.00
	109_C	0.419	0.00	0.533	-0.846
110	110_B	0.410	0.00	1.00	0.00
	110_C	0.410	0.00	0.522	-0.853
111	111_B	0.328	0.734	0.00	0.679
	111_C	0.328	0.707	0.00	0.707
112	112_B	0.337	0.00	0.565	-0.825
	112_C	0.337	0.00	0.957	-0.290
113	113_B	0.337	0.00	0.565	-0.825
	113_C	0.337	0.00	0.957	-0.290
114	114_B	0.333	0.829	0.508	0.235
	114_C	0.333	0.936	0.348	-0.052

115	115_B	0.340	0.00	0.649	-0.761
	115_C	0.340	0.00	0.998	0.054
116	116_B	0.336	0.00	1.00	0.00
	116_C	0.336	0.936	0.348	-0.052
117	117_B	0.17	0.00	1.00	0.00
	117_C	0.17	0.00	0.741	-0.671
118	118_B	0.301	0.883	0.00	0.47
	118_C	0.301	0.734	0.00	0.679
120	120_B	0.328	-0.883	0.00	-0.470
	120_C	0.328	0.883	0.00	0.470
122	122_B	0.415	-1.00	0.00	0.00
	122_C	0.415	0.998	-0.05	0.027
123	123_B	0.2422	0.00	1.00	0.00
	123_C	0.299	0.00	0.966	-0.08
124	124_B	0.2422	0.00	1.00	0.00
	124_C	0.299	0.00	0.966	-0.08
125	125_B	0.5756	0.883	0.00	0.470
	125_C	0.360	0.00	1.00	0.00
126	126_B	0.33	0.998	-0.05	0.027
	126_C	0.33	0.840	0.528	-0.136
129	129_B	0.172	$T_{X124}(0.122)$	$T_{Y124}(0.122)$	$T_{Z124}(0.122)$
	129_C	0.608	0.00	0.919	-0.394
130	130_B	0.326	0.840	0.528	-0.136
	130_C	0.326	0.00	0.919	-0.394

Table A2.1 Curves and Their Intermediate Control Points

* α -- Cantle angle

** β -- Seat angle

*** γ -- Dome angle

\# $T_{x133}(1.0)$ --- x component of the tangent of curve 133 at u=1.0

Due to symmetry, only half of the curves, shown in Fig. A1.1 and Fig. A1.2, are calculated. All curves on the other side are calculated by reflecting these curves.

A2.2 Blending Curves

These curves forms the boundaries for the blending surfaces between the upper side and lower side surface patches. All such blending curves are as shown in Fig. A2.1. All blending curves are calculated using the following relation:

$$\boldsymbol{P}_{intermediate} = \boldsymbol{P}_{end} \pm k \times \mathbf{v},$$

Where,

$\boldsymbol{P}_{intermediate}$ = the intermediate control point of the blending curve,

$\boldsymbol{P}_{end}$ = Nearest endpoint of the blending curve,

k = constant,

$$\mathbf{v} = \frac{\left(P_{conti_{intermediate}} - P_{end}\right)}{\left|P_{conti_{end1}} - P_{conti_{end2}}\right|}$$

where, $P_{conti_{intermediate}}$ = the nearest intermediate control point of the curve with which the considered blending curve is C^1 continuous.

$P_{conti_{end1}}$ and $P_{conti_{end2}}$ = the end points of the curve with which the considered blending curve is C^1 continuous.

The information needed for calculating the blending curves is given in Table A2.2 below.

Blending Curve	Control Point	C^1 Continuous Curve	K
36	37B	6	0.443
	36C	39	0.443
37	37B	9	0.443
	37C	42	0.443
71	71B	0	0.715
	71C	60	0.715
135	135B	13	0.428
	135C	104	0.428
136	136B	14	0.344
	136C	79	0.344
140	140B	54	0.500
	140C	101	0.500
144	144B	46	0.500
	144C	113	0.500
148	148B	43	0.500
	148C	115	0.500
154	154B	28	0.500
	154C	74	0.500
157	157B	68	0.500
	157C	124	0.500
162	162B	60	0.500
	162C	125	0.500

Table A2.2 Blending Curves and Their Intermediate Control Points

Curve and Surface Manipulations

A3.1 Subdivision of a B-Spline Curve

The subdivision of a B-spline curve is achieved as a by-product of the De Casteljau Algorithm. The algorithm is

$$b_i^r(u) = (1-u)b_i^{r-1}(u) + ub_{i+1}^{r-1}(u) \qquad \textit{for } r = 1, \ldots, n;\ i = 0, \ldots, n\text{-}r$$

where, u = parameter value.

Let,

$\{V_i^a : i = 0, \ldots, n\}$ be the control points of the first segment $\mathbf{r}^a(t)$, and

$\{V_i^b : i = 0, \ldots, n\}$ be the control points of the first segment $\mathbf{r}^b(t)$.

then they are obtained from the De Casteljau points as

$$V_i^a = b_0^i(u) \quad \textit{for } i = 0, 1, .., n,$$

$$V_i^b = b_i^{n-i}(u) \quad \textit{for } i = 0, 1, .., n$$

A3.2 Tightening of Control Net

As stated in Section 4.1, for solving the problems of continuity in the region near the symmetric plane, most of curves are represented by six control points, or in other words, their control net is tightened. For calculating those six control vertices from the initial four vertices, the OSLO algorithm is used:

OSLO algorithm:

Consider a curve p(t) defined by n control points

Then,

$$\mathbf{P}(t)=\sum_{i=1}^{n+1} B_i N_{i,k}(t)$$

The new curve R(s) with m control points can be defined by

$$\mathbf{R}(s)=\sum_{j=1}^{m+1} C_j M_{j,k}(s) \qquad \text{Where } m > n$$

The objective is to define the new defined polygon vertices Cj such that P (t)=R (s). through OSLO algorithm the new C_j's are

$$C_j=\sum_{i=1}^{n+1} \alpha_{i,j}^{k} B_i \qquad 1 \le i \le n, 1 \le j \le m$$

Where $\alpha_{i,j}^{k}$'s are given by the recursive relation

$$\alpha_{i,j}^{1}=\begin{cases} 1 \rightarrow x_i \le y_j < x_{i+1} \\ 0 \rightarrow otherwise \end{cases}$$

$$\alpha_{i,j}^{k}=\frac{y_{j+k-1}-x_i}{x_{i+k-1}-x_i}\alpha_{i,j}^{k-1}+\frac{x_{i+k}-y_{j+k-1}}{x_{i+k}-x_{i+1}}\alpha_{i+1,j}^{k-1} \qquad \text{Where } \sum_{i=0}^{n+1}\alpha_{i,j}^{k}=1$$

A3.3 C^1 Continuity between Adjacent Surfaces

The two adjacent surface patches are said to be C^1 *continuous* if the tangent planes of the two patches along the common boundary curve coincide. Hence a simple algorithm is used here to make the two adjacent surface patches *C1 continuous*. Fig. A3.1 explains the meaning of terms used.

Algorithm:

1) Calculate the normal vector n for the tangent plane.

$$\boldsymbol{n} = (\mathbf{P}_{10} - \mathbf{P}_{00}) \,.\, (\mathbf{P}_{01} - \mathbf{P}_{00})$$

2) Find the deviation h of the intermediate control point in consideration and the tangent plane. This deviation h may be positive or negative.

$$h = (\mathbf{P}_{11} - \mathbf{P}_{00}) \,.\, \boldsymbol{n}$$

3) Then find the corrected control point using following relation.

$$\mathbf{P}_{11}' = \mathbf{P}_{11} + h\,\boldsymbol{n},$$

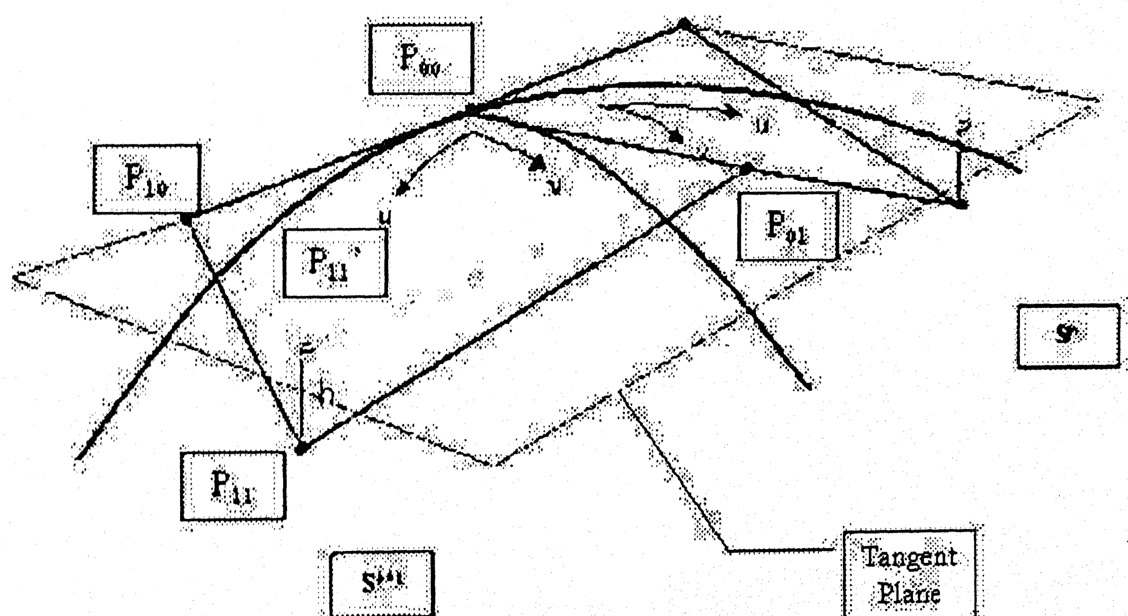


Fig. A3.1 Adjacent Surface Patches and Tangent Continuity